

IDENTIFICATION

PRODUCT CODE: MAINDEC-12-D3AB-D
PRODUCT NAME: PDP-12 TAPE CONTROL TEST (PART 1 of 2)
DATE CREATED: AUGUST 12, 1970
MAINTAINER: DIAGNOSTICS GROUP
AUTHOR: JAMES KELLY

1. ABSTRACT

The tape control diagnostic is designed to completely test the TC-12 tape control logic on a gate by gate basis. This program requires that at least one (1) Linc tape transport be connected and that the computer be thoroughly debugged. All communications between the program and the operator are via the teletype. See appendix A for a detailed description of the tests performed by part 1 of this program.

2. REQUIREMENTS

2.1 Equipment

- 1) A standard basis PDP-12
- 2) A TC-12, PDP-12 linc-tape controller.
- 3) At least one linc-tape transport.
- 4) A ASR-33 teletype or equivalent.

2.2 Storage

This program is designed to run in memory bank 0 only and it occupies virtually all bank 0 not occupied by the Binary and Rim Loaders.

2.3 Preliminary Programs

All PDP-8 and 12 mode basic instruction diagnostics and exercisers must have been successfully run prior to running the program.

3. LOADING PROCEDURE

3.1 Method

This program must be loaded with the binary loader. If you are unfamiliar with the proper binary loading procedures refer to Appendix A of this document, otherwise proceed with the following:

- a) Set the teletype reader switch to FREE.
- b) Open the teletype reader and insert the program tape so that the arrows on the tape are visible to and pointing toward the operator.
- c) Close the reader and set the reader switch to START.
- d) Set the teletype front panel switch to START.
- e) Set the left switches to 7777.
- f) Set the right switches to 4000.
- g) Set the mode switch to 8 mode.
- h) Depress I/O preset.
- i) Depress start LS.
- j) When the program tape has been read in the ACCUMULATOR must be 0000, if it is not, a read-in error has occurred and one might try reloading the binary loader.
See Appendix A.
- k) Remove the program tape from the reader.

4. STARTING PROCEDURES

This preliminary set up procedure is critical and any omission will result in an error.

- a) Set on e and only one of the Linc tape transport number thumbwheels to 0 - on some transports the 0 position is actually represented by 8.
- b) Set all other transports, if available, to all different numbers i.e. no two transports have the same unit number.
- c) Set the WRITE ENABLE switch on every transport to the enabled condition.
- d) Set all transport switches to REMOTE.
- e) Remove any Linc tapes currently on the transport.
- f) Set the left switches to 0200.

g) Set the right switches to 0000.

NOTE: Setting the right switches to configurations other than 0000 will cause the computer to scope loop, halt on error, and perform other options. These options are discussed later.

h) Set the MODE switch to 8 mode.

i) Depress I/O preset.

j) Depress START LS.

The program is running.

4.1 Control Switch Settings

A set of 6 optional mode switches consisting of right switches 0-5 has been included for the convenience of the test engineer, they are:

SR00 = 1 suppress error halts
SR01 = 1 suppress type outs ring bell on error
SR02 = 1 scope loop on a failing test
SR03 = 1 scope loop on a non-failing test
SR04 = 1 unit does not have extended tape fields
SR05 = 1 suppress the bell

The switches have an order to precedence associated with them, for example, if the option switches were set so as to cause a typeout and an error halt, it is obvious that the typeout must precede the halt. Right switch 00 set to a one will prevent the computer from halting when an error is detected. Depending on the condition of the other switches, we may:

- 1) Go on to the next test after typing out a message
- 2) Stay in the same test.

Right switch 01, if set to a one, will prevent the computer from typing anything out and instead causes a bell to ring at every detected error. The purpose of this is merely to inform the operator that an error has occurred and nothing more.

Right switch 02, if set to a one, will prevent the computer from exiting the current failing test.

Using switches 00, 01, 02 several use-ful combinations of error analysis present themselves:

SR0	SR1	SR2	Explanation
0	0	0	Type error data and halt
0	1	0	Ring bell and halt
1	0	0	Continuous typing of data with no halt
1	1	0	Ring bell at every failure
1	0	1	Continuous typing of data in current test

In general, an error halt is useful for scoping status of the machine immediately following an error.

Right switch 03, if set to a one, will cause the program to remain in the current test, so long as no failures occur.

Right switch 04, if set to a one, will cause the program not to test the extended tape field register. (4K System)

Right switch 05, if set to a one, will inhibit the bell from ringing. Under normal operation the program will ring the bell about once every 1 and 1/2 minutes.

5. MESSAGE FORMAT

- 1) The message format is designed to yeild the maximum possible information with the minimum amount of typing. To that end the following format has been selected as the best of both worlds; i.e., amount of information vs. the amount of typing.

Example:

LTR AC RWB (VIA TB) Failed

AC RWB

0001 0000

0002 0000

- LTR 2) The first item typed, in this case LTR, refers to the logic page on which the logic which is being tested is drawn. In this case the message tells us that the logic under test is located in the (LTR) Linc Tape Register Logic. It should be understood that the trouble is associated with the Linc tape register logic but not necessarily on the page referenced. For example on this test the data transfered may not have gotten from the computer to the tape control or it may not have been read back properly. Both of these problems would cause a typeout such as this, indicating a bad register, when in fact the trouble was in getting data to or from a register.

- AC RWB 3) The second times typed, AC RWB, indicate the two registers involved; the Ac, referring to the tape control read write buffer.

- (VIA TB) 4) The third item, enclosed in brackets, is a modifier. In this example the program is capable of loading the RWB in

one of three (3) ways, from the tape transport, from the AC serially via a maintenance gate, and from the AC via the TB. Obviously, three different trouble shooting techniques are required depending on which of the three data paths are bad. The modifier in this case points out which one of three data paths failed.

FAILED 5) The word "failed" is typed to be sure the operator understands that this message indicates trouble and is not interpreted as a status report.

AC RWB 6) The two registers whose data are shown are named on the second line of the typeout to ensure that the operator is aware that the data type-out is AC and RWB and not the TB. Usually the registers involved do not have their names typed out unless there is a possibility of confusion.

0001 0000 7) The numerical data type outs are also in a special format.

When more than one number is typed the first number is always the source number. In this typeout the 0001 is the number which was in the AC prior to the transfer. The 0000 is the number in the AC after the contents of the RWB were read back into the AC. An engineer must always refer to this document and locate the exact type in order to properly interpret any message or data typeout.

6. MONITORS

This program contains two monitors, an error monitor and a non-error monitor. The error monitor handles scope looping on errors, message typeouts, and determines what data shall be used in a failing test. The non-error monitor is an extremely simple subroutine whose only function is to allow a test to loop continuously even when no error exists.

The following example will be used to illustrate a typical coding sequence, involving the comparing of a true number with the actual results of an operation.

1)	TESTX,	TAD	REGB	/Fetch Received data
2)		CIA		/Negate
3)		TAD	REGA	/Subtract from sent number
4)		SNA	CLA	/Were they equal
5)		JMS	1 NERROR	/AC was zero, they were equal
6)		JMS	1 ERROR	/Sent Data Received data unequal
7)		GOOF		/Message TAG
8)		HLT		/Error HALT
9)		SKP CIA		/EXIT
10)		TESTX		/Scope Loop Pointer

The number shown in the left margin are for reference purposes only and are not part of the coding.

The first three lines performed in order (1) fetch the resultant of the test from "REGB" this could have been any memory register or any hardware register which can be read under computer control.

The test data is converted to twos' complement form (2) and subtracted from the correct results (3) The test of the data

(i.e. were they equal) takes place on line (4) and based on this test we go either to the non-error (NERROR) subroutine or if the ACCUMULATOR is not zero the error (ERROR) monitor. It should be obvious that any decision making instruction can be used to ascertain which monitor we hand control to.

Lines (5), (6), are the actual monitor calling instructions. Line (7) (GOOF) is the address of the first memory location of the error message which will be typed out in case of a failure. Line (8) is error halt. If error should occur, and the switches are set so as to allow an error halt, this address will be the one at which it will halt.

Line (9) is an exit. If an error halt occurs, pressing 'continue' will cause the computer to execute the skip and exit to the next test.

Line (10) contains the address at which this test is begun again. For example, after completing one pass thru this test routine the monitors will execute an effective jump Indirect the contents line (10) and redo this test routine.

6.1 Non Error Monitor

The non error monitor has two functions. The first is to increment "REGA". "REGA" is a common tally register used to count 4096 passes thru a test and to notify the non-error monitor via an "ISZ" loop when the required number of passes have been performed and thereby causing an exit. The second function is to examine RSW 03 and if set, inhibit exiting to the next test.

In some tests, particularly those associated with time delays or mechanical delays, it becomes prohibitive to make 4096 passes thru a test. To circumvent this, it is possible to preset "REGA" to "7777" so as to make only a single pass thru a test, or any number of passes from 1 through 4096.

6.2 Error Monitor

The error monitor is the major monitor responsible for all modes of communicating errors to the operator. The usage of switch inputs has been completely discussed under part 4.1 control switch settings and will not be discussed here. Several salient features of the error monitor are as follows. The first "scope loop on failing test" (SR02=1), is designed to cause the monitor to inhibit incrementation of "REGA", and to inhibit the advance of the random number generator. An example of its use might be in testing any of the 12 bit registers. Assume that bit 00 can never be set to a one because of some as yet unknown hardware error. This malfunction will become known the first time the number 4000 is loaded into it because the read back will show 0000 - normally the next number to be tried will be 4001, 4002 etc. with each being typed out and each causing different data to be transferred. To facilitate scope testing of this problem we must eliminate type outs and prevent the data from changing. This is easily accomplished as explained under switch settings.

An error message is always formatted such that all of the non numeric characters are typed out first followed by the numerics. The contents of some memory register, other than those selected by the programmer, may be of interest to the field engineer. For

in a random data transfer test it is impossible to determine the number of successful data transfers, because only the errors are typed out. Let's presume that the engineer wishes to type out the pass counter i.e. "REGA" memory address 0004. It is necessary to modify the message type-out string as follows:

	BEFORE		AFTER
GOOF,	0001 0001	GOOF,	0001 0001
	0203 0203		0203 0203
	EXITA 7777		EXITA 7777
	REGB 0005		REGB 0005
	REGC 0006		REGC 0006
	EXIT 0000		REGA 0004
			EXIT 0000

The following shows the before and after type out.

ABC 7351 7350	ABC 6773 6253 0037
---------------	--------------------

The type out on the right shows the contents of the pass counter and will indicate if all random numbers failed or if only some of them failed. It is absolutely necessary to restore the toggled in modifications to the message type out in order to prevent erroneous type outs in other messages.

7. MAINTENANCE INSTRUCTION SET

MNEMONIC	CODE	MODE	OPERATION
LMR	6151	PDP-8	Load maintenance register
AC			
0			TO MAINT INST REG
1			TO MAINT INST REG

- 2 TO MAINT INST REG
- 3 TO MAINT INST REG
- 4 CLEAR TAPE DONE FLOP
- 5 SKIP ON TAPE DONE
- 6 GENERATE A SIMULATED TFO, TT1, TT2 PULSE
- 7 GENERATE A SIMULATED TT3, TT4, PULSE
- 8 SIMULATE MARK INPUT
- 9 SIMULATE DATA 1 INPUT
- 10 SIMULATE DATA 2 INPUT
- 11 SIMULATE DATA 3 INPUT

Bits 0, 1, 2 3

The contents of ACCUMULATOR bits 0, 1, 2, 3, are loaded as a command into the maintenance instruction register. The command will be executed if and only if the XFR IOT (6154) is generated; the function of these commands are discussed later.

Bit 4

Executing the LMR command with AC bit 4 set will unconditionally clear the tape done flag.

Bit 5

Executing the LMR command with AC bit 5 set will cause the computer to skip the next instruction in sequence if the tape done flip-flop was set.

Example:	CLA CLL	/Clear AC, L
	TAD K0100	/Set Bit 5
	LMR	/IOT 6151
	HLT	/Tape done was zero
	HLT	/Tape done was one

It should be noted that these commands are not designed to be microprogrammed; for example, setting both Bit 4 and 5 and executing LMR in an attempt to SKIP and clear on the tape done flag is unwise.

Bit 6

Executing the LMR command with AC bit 6 set will generate in order the timing pulses TT0, TT1, TT2 regardless of the state of the tape control logic.

Bit 7

Executing the instruction LMR with bit 7 set will generate in order TT3 and TT4. It should be noted that to generate an entire timing stream consisting of pulses TT0, TT1, TT2, TT3, and TT4, it is necessary to generate TT0, TT1, and TT2 first followed by a second command to generate TT3 and TT4.

Bit 8

Is used to simulate an input to the mark window, see bits 9, 10, 11 below.

Bits 9, 10, 11

Executing the instruction LMR with data in AC bits 9, 10, 11 will simulate reading data off the data heads into the Read-write buffer. This feature is useful in testing the tape control without a transport.

TRC 6152 PDP-8 Tape Register Clock

AC

0	Generate tape preset
1	Shift RWB once left with no end around carry
2	Transfer TB to RWB

TRC 6152	PDP-8	Tape Register Clock
3		Add TB and TAC place the results in TAC
4		Ø Tape word flip-flop
5		Set Forward
6		Set unit 1
7		Set Backward
8		Set write sync flip-flop
9		Set 8 mode tape motion
1Ø		Select 8 tape mode
11		AC11→LTP8 Write if AC10 is set

The tape register control command TRC (6152) in conjunction with selected bits in the AC can perform a number of direct non-conditional operations. Several of the commands are useful primarily for logic testing. The rest, although they are tested, are used in normal machine operation.

Bit Ø:

Executing the command tape register control TMR (6152) in conjunction with AC bit Ø set to a one generates the internal signal tape preset. Tape preset, in general, sets all control flops to a null state, which may be either a one or a zero.

Bit 1:

TMR in conjunction with AC bit 1 will cause the Read-write buffer (RWB) to be shifted once to the left, observing it from the computer accumulator, or once up on the logic drawings. Data shifted out of RWB is lost and a logic Ø is shifted into the low order bit.

Bit 2:

TMR in conjunction with AC Bit 2 will cause the contents of the TB to be copied into the RWB. The previous contents of the RWB are lost, the TB remains unchanged. It should be noted that the only path by which the RWB may be loaded in parallel is via the TB register.

Bit 3:

AC bit 3 causes the contents of the TB register to be added to the contents of the TAC register, with the resultant being retained in the TAC.

Example:

BEFORE	AFTER
TB = 7321	TB = 7321
TAC = 0412	TAC = 7733

Bit 4:

Clear the tape word flip-flop. This bit generates an unconditional clear.

Bit 5:

Set forward. The direction flip-flop is set to a one unconditionally. This command is useful for diagnostics

Bit 6:

Set unit 1. When executed, this command will select tape transport unit 1. It should be noted that this command can only select unit 1 if the extended transport select register (part of extended operations register bits 10, 11) is set to a zero. If the extended transport select register is not zero, the appropriate odd numbered unit is selected.

Bit 7:

Set Backward the direction flip-flop is set to zero unconditionally. This command is useful for diagnostics.

Bit 8:

Set write sync. This command unconditionally sets the write-sync flip-flop.

Bit 9:

Set 8 tape motion is used with the TC12-F option. This option allows the reading and writing of tapes written in PDP-8 DECTape format.

Bit 10:

Set 8 write selects 8 tape mode.

Bit 11:

Set LTP8 write, AC bit 11 is provided as a data input to the 8 tape write flip-flop. If AC bit 10 is set, the bit is clocked into the flop.

XFR 6154 PDP-8 Transfer

Maintenance register

00	AC to TB
01	AC to TBN
10	AC to TAC
11	AC to TMA
20	TMA setup to AC
21	TBN to AC
30	TB to AC
31	RWB to AC
40	Mark window to AC

41	States to AC
50	Units and motion to AC
51	Tape instruction register to AC
60	Misc status 1 to AC
61	Misc status 2 to AC
70	TMA to AC
71	Unused, returns all zeros to AC

In general, all data transfers into the AC using maintenance commands are 1's transfers, that is, they do not clear the AC prior to inserting data. All data transfers from the AC into tape control register are jam transfers. Any data which was in a tape control register is lost.

00 AC to TB

The current contents of the AC are transferred into the TB (Tape Buffer). The AC data is unaffected, the previous contents of the TB are lost.

01 AC to TBN

The current contents of the AC are transferred into the TBN (Tape Block Number). The AC is unaffected, the previous contents of the TBN are lost.

10 AC to TAC

The current contents of the AC are transferred into the TAC (Tape Accumulator). The AC is unaffected, the previous contents of the TAC are lost.

11 AC to TMA

The contents of the AC are transferred into the TMA (Tape Memory Address). The AC is unaffected the previous contents of the TMA are lost.

20 TMA setup to AC

The contents of the TMA setup register are "ORED" into the AC.

The contents of the TMA setup register are unaffected.

21 TBN to AC

The contents of the TBN register are "ORED" into the AC.

The contents of the TBN are unaffected.

30 TB to AC

The contents of the TB register are "ORED" into the AC.

The contents of the TB are unaffected.

31 RWB to AC

The contents of the RWB (Read write buffer) are "ORED" into the AC. The contents of the RWB are unaffected.

40 Mark Window to AC

The contents of the tape mark window and its associated mark decoding gates are "ORED" into the AC. The contents of the Mark window register are unaffected. The data format is as follows:

AC

00	Window Shade
01	Window 00
02	Window 01
03	Window 02
04	Window 03
05	End Zone mark
06	Check mark
07	Guard mark
08	Data mark
09	Final mark
10	Block mark
11	Intermediate zone mark

41 States to AC

The contents of several flops and the levels of several gates are "ORED" into the AC. Reading the data does not affect its state. The data format is shown below:

AC

00	TAC = 7777
01	IDLE Mode = 1
02	Search mode = 1
03	Block mode = 1
04	Check word mode = 1
05	Turn around mode = 1
06	Write Flop = 1
07	Write cycle flop = 1
08	Acip delay not timed out
09	Tape timing OK
10	Timing OK gate set
11	Tape fail delay

50 Units + MTN to AC

The data concerning transport selection, motion, direction, unit select, and write enable is "ORED" into the AC.

AC

00	UNIT 0	Selected
01	UNIT 1	Selected
02	UNIT 2	Selected

03	UNIT 3	Selected
04	UNIT 4	Selected
05	UNIT 5	Selected
06	UNIT 6	Selected
07	UNIT 7	Selected
08	MOTION FLOP	(1)
09	DIRECTION FLOP	(1)
10	UNIT SELECT	
11	WRITE ENABLE	

Bit 10 Unit Select

This bit indicates that one and only one unit is selected at a time.

51 TINST to AC

The contents of the tape instructions register decoder, the 1 bit and the group register are transferred to the AC. The contents of these datum are unaffected; however, the previous contents of the AC are lost.

AC	
00	RDC
01	RCG
02	RDE
03	MTB
04	WRC
05	WCG
06	WRI
07	CHK
08	I BIT

09	GP 0 (1)
10	GP 1 (1)
11	GP 2 (1)

60 Misc Status 1 to AC

This command transfers the status of several important levels to the AC. The format is shown below:

AC

00	PHASE GATE
01	PROGRESS FLOP (1)
02	LC 00 (1) Line Counter
03	LC 01 (1) Line Counter
04	MARK CHANNEL WRITE
05	DATA CHANNEL 1
06	DATA CHANNEL 2
07	DATA CHANNEL 3
08	GP CNT = GP FLOP
09	GP CNT 0 (1)
10	GP CNT 1 (1)
11	GP CNT 2 (2)

61 Misc Status 2 to AC

Bit 0 of the AC is set to a one if the LTP8 tape select flip-flop is set.

70 TMA to AC

The contents of the TMA register are transferred to the AC. This contents of the TMA are unchanged, the original contents of the AC are lost.

71 Unused.

General Information

This program will test the PDP-12 tape control logic on a gate by gate basis. This was made possible by including a 16 instruction maintenance register in the basic design. The line tape maintenance register (LTMR) facilitates examination of all major registers and the majority of status bits, control levels, etc. associated with the tape control. It is usually possible to isolate the fault to no more than one or two modules by analyzing the diagnostic type out and referring to the appropriate logic diagram.

This program is written in several major and minor sections designed to point to a failure on an appropriate page in the logic drawings.

The four major sections in order are:

- 1) Test out, in so far as possible, the maintenance logic. This includes the maintenance mode flop, maintenance instruction register, tape preset and the tape buffer to computer accumulator (TB to AC) data read back portion. This group of tests are a necessary preliminary preliminary diagnosis in order to be reasonably certain that the following tests fail because of logic failures and not because of failures in the maintenance instruction register.

- 2) Test out the 12 bit registers TAC, TH, TMA setup, RWB and TBN. Two tests are performed on each register; the first test is a binary up count sequence; the second a random number sequence. The binary sequence ascertains that all flops can be both set to a one and set to a zero. In addition it also proves mutual independence of the data paths i.e. the flops can move independently of one another and the input and output data paths are not shorted in any way. The random number test causes each bit of the registers to toggle at a relatively high speed, in contrast to the binary sequence wherein only the low order bits toggle at

high speed. A second useful effect of the random number sequence is that it leaves the tested register in a random state, a technique sometimes useful for discovering grossly illogical wiring errors.

3) Test all minor registers; i.e., registers with less than 12 bits. These include unit select registers, mark window, extended operations register, etc. In this series of tests an attempt is made to diagnose all flops and all sequenceable flops such as the major state generator. In most cases random number tests are not performed due to the fact that an unknown state in any control flop might have an adverse affect on the succeeding tests.

4) Test all gates using as many as possible input combinations. For example, AND gates are first tested with all inputs true to determine if the gate will function. Next, each individual input is set false in turn to see if each input is expressed in the output. OR gates are tested by allowing at least one input and maybe more to become true and monitoring the output for a true condition. This is followed by setting all inputs to false, monitoring the output for false, followed by setting each input in turn to true and checking the output.

A significant number of gates in the tape control can not be directly tested; tape state logic TC12-Ø-LTS, for example. These untestable gates are evaluated indirectly by logical deduction; i.e., all testable inputs to these gates are tested as inputs to other gates. An example: Block mark (BM) is a functional input to the LTS Logic. As previously stated, it is extremely difficult to prove that a correct (BM) appears at the LTS Logic,

however, it is easy to prove that the (BM) signal itself is functional, that other gates using the (BM) signal function, and that the (BM) signal is not shorted, either to logical low or logical high. The technique used to circumvent the apparent difficulty in testing multilevel logic is as follows:

- a) Attempt to set all inputs to a group of logic to a state which will cause a true output at some software observable point.
- b) Attempt to disqualify the output by causing one input at a time to change to a state which will cause the output to become false.
- c) In many cases a gate will be qualified or disqualified, as the case may be for a specific period of time regardless of what the diagnostic program does. An example of this is any gate whose inputs are a function of tape delays.

NOTE:

These print-outs should not be regarded as accurate definitions of the problem.

A second print-out can usually be obtained by pressing "continue" with RSW 00 and 01 set for a few seconds to move off that particular test, then restoring the RSW to await subsequent failures.

This may point to a more basic approach to the problem.

The following seven (7) error messages are associated with tests of the
TAPE MAINTENANCE REG. TC12-0-LTMR:

LTMR MAINT MODE FLOP
0040 0000

A Transfer of data from AC06 to the Maint Mode Flip-Flop was attempted and failed. The first number represents the number in AC bit 6 represents the status of the Maint Flop after the transfer. The data transfer was done using the Linc Mode Instructions AXO 0001 and XOA 0021.

LTMR MAINT MODE TAPE PRESET
0040 0040

The contents of bit 6 in the left most word was transferred to the Maint Mode Flop; a tape preset signal was generated in an attempt to clear the Maint Mode Flop; the second number indicates the contents of the Maint Flop after the tape preset.

LTMR MAINT IR TAPE PRESET
4000 0005 0004

The three numbers typed out represent in order: the number transferred to the Maint IR, the number transferred to the TB, and third the number received from the TB. The test on the Maint IR is performed as follows: a number consisting of a 12 bit binary upcount sequence is transferred into the TB register; a second number, consisting of Bits 0, 1, 2, 3 of the first number typed out, is transferred into the Maint IR register; the tape control signal "tape preset" is then generated under

program control. This signal should clear the Maint IR. Since the Maint IR command consisting of all zeros is a command to transfer the contents of the AC into the TB, an attempt is made to do so. Next, the program reads the contents of the TB to ascertain if it contains the data which was transferred.

LTMR IOT

6152

0000

The test of IOT 6152 Decoder is accomplished by setting the Maint Mode Flop and attempting to clear it by executing 6150 and 6552 IOTs with AC00 set. (A "tape preset" is executed by setting AC00 and executing a 6152). AC06 of the typed out number indicates the state of the Maint Flop.

LTMR IOT

6154

0000

IOT 6154 is the IOT used to execute the instruction currently contained in the Maint IR. This routine sets the TB register to 5252, then checks the IOT Decoder by executing 6150 and 6554 IOTs with 7777 in the AC.

LTMR IOT

6151

0000

The IOT 6151 is used to Load commands into the Maint IR. The Decoder is tested by loading the command "TB to AC" into the Maint IR, then attempting to reload a 0000 command over it, using false 6XX1 commands.

L T M R I R R E G F A I L E D T O S E T

0000

An attempt was made to set the Maint IR to 17
i.e., all Bits set. This command is unused and
should return all zeros to the accumulator
when executed. If all bits in the Maint IR did
not get set the information, Read Back will be
non-zero and will be typed out.

The following fifty-nine (59) messages are associated with tests of the LTRA Bits 00-11 (TC12-0-LTRA, B, C, D, E, F.) It should be noted that each of these error messages can be produced by two (2) different tests: the binary sequence test, and the random sequence test. The random test also contains a test (in some cases) to detect load or read commands that also respond to false conditions.

LTR TB AC FAILED

AC TB

0000 0000

A data transfer between the computer accumulator and the tape control tape buffer failed.

LTR TBN AC FAILED

AC TBN

0000 0000

A data transfer between the computer accumulator and the tape control tape buffer failed.

LTR TAC AC FAILED

AC TAC

0000 0000

A data transfer between the computer accumulator and the tape control accumulator failed.

LTR TMA SETUP AC FAILED

AC TMA SETUP

A Data transfer between the computer accumulator and the tape memory address setup register failed.

LTR TMA AC FAILED

AC TMA

0000 0000

A data transfer between the computer accumulator and the tape memory address register failed.

LTR AC RWB (VIA TB) FAILED

AC RWB

0000 0000

A data transfer between the computer accumulator and the Read/Write buffer failed. It should be noted that the actual path of data transfer is from the accumulator to the TB, from the TB to the RWB, and from the RWB to the AC. The data transferred into the TB has been tested previously so it can be presumed that any errors in this test are the result of the TB to RWB transfer or the RWB to AC transfer.

LTR RWB SHIFT

0000 0000

A number was transferred to the RWB, the RWB was shifted 1 place and the results read back. The first number typed is the correct data after the shift. The second number is the actual data as read from the RWB.

LTR AC RWB (VIA MAINT GATE) FAILED
AC RWB
0000 0000

A 12 bit word was loaded into the RWB via the 3 Maintenance inputs serially, and transferred to the AC in parallel, failed.

LRE MB TBN FAILED (EXT ADD=0)
0000 0000

A data transfer between the computer memory buffer MB and the tape control tape block number TBN failed. In this case, the Extended address flip-flop was 0: therefore, no transfer into Bits 0, 1, 2 can take place.

LRE MB TBN FAILED (EXT ADD=1)
0000 0000

A data transfer between MB and TBN failed; in this case, the extended addressing flop was set so that a full 12 bit transfer should take place.

LTR TB TAC ADD FAILED
TB TAC GOOD ACTUAL
3777 4000 7777 4777

This message can be produced by one of three diagnostic subroutines. The first floats a one thru the TB and a zero thru the TAC. The second does the inverse. The third adds two random numbers. The first and second numbers are the two arguments. The third number "GOOD" represents the answer as computed by the computer. The fourth number is the answer as computed by the logic under test, and by comparing the "GOOD" versus the "ACTUAL" it is usually

possible to determine the source of trouble.

LCX EXTEND OPS DATA XFER
0000 0000

A transfer of data between the AC and the extended operations register failed. It should be noted that bits 0, 1, 2 represent the Tape Fields and will only be tested if SR04 is set to a zero prior to this test.

LCX EXTEND OPS PRESET
0000

All bits in the extended operations register are set. A tape preset signal is generated to clear the register, after which it is read and tested; if the resulting state of the register is not zero, that number is typed out.

LCX LOAD EXT OP REG
0000

This test checks the decoder for the command "AOX" by executing commands other than 0001. This test is performed by setting all bits of the extended ops register to 1's and then attempting to load it to zero using false AOX commands. (0016 & 0301)

LCX EXTEND OPS READ BACK
0000 0000

This test checks the decoder for the command "XOA". The extended ops register is set to all ones followed by an attempt to read it back using false XOA commands. (0016, 00321, 0700)

LCX LDF 03 DATA XFER FAILED
DF TDF
0000 0000

An attempt was made to transfer the contents of the computer "DATA FIELD" into the "TAPE DATA FIELD". The contents of the Tape data field are read back and compared with the data field register; any errors are printed out.

LCX LIF 03 DATA XFER FAILED
IF IF
0000 0000

An attempt was made to transfer the contents of the "INSTRUCTION FIELD" into the "TAPE INSTRUCTION FIELD". Setting the instruction field in the processor is accomplished by handing the program off to four different subroutines in each of the four memory quarters.

LIN TAC EQ 7777 (TAC=7777)
7777

This type out can be caused by one of three diagnostic subroutines. The first is a test to be certain that the "TAC=7777" gate can detect the condition whereby the tape accumulator is equal to all ones. The second subroutine will float a single zero bit thru the TAC and test the "TAC=7777" gate. The final test runs a stream of random (non 7777) numbers into the "TAC" and tests the output of the TAC=7777 gate.

LIN TINR TAPE PRESET
0000 0000

A number shown as the first number typed was loaded into the tape instruction register, tape preset was generated, and the state of the instruction register tested to ascertain if it had been zeroed. It should be noted that the first number typed is the binary number loaded into the instruction register.

The second number represents the output of the decoder.
(See IOT 6154/MAINT IR=51-TINST to AC)

LIN INSTRUCTION DECODER
0000 0000

An instruction register test has failed. The first number is the binary number loaded into the instruction register. The second number represents the output of the decoder. Reference should be made to the maintenance instructions for a complete explanation.

LIN I F BIT
0000 0000

The I bit was loaded with the data shown in bit 7 of the first number typed; the data in bit 7 of the second word is the data transferred back to the AC.

LTC UNIT SELECT
0000 0000 0000

A test was made of the unit select gates to determine if each unit could be selected. This test is not dependent on the actual number of transports connected. The first number typed indicates the actual unit we were trying to select. The second number typed shows what output of the unit select decoders should be and does not depend on the decoders working. The last number typed is the actual output of the decoders. By comparing the actual and computed output the engineer can determine what is wrong.

LTC UNIT SELECT PRESET
0000

The unit select register was set to 7. Tape preset attempted to zero it; the error number is printed out.

LTC UNIT 1 FAILED
0000

The command, "SET UNIT 1" was given. The unit selects were read back to see if unit 1 was in fact selected.

LTC UNIT SEL FAILED
0000

The logic which detects that one and only one unit is selected has failed; bit 10 shows the status of the single unit select Bit.

LTC WRITE ENABLE FAILED

0000

The write enable switch on the transport selected is not set, or the signal is not returning to the tape logic.

LWN MARK WINDOW DIR (0)

0000 0000

The data in bits 08, 09, 10, 11 of the first number typed were transferred into the mark window and read back. The second word (bits 08, 09, 10, 11) show the read-back data; the direction flop was in the backward direction.

LWN MARK WINDOW DIR (1)

0000 0000

The data in bits 08, 09, 10, 11 of the first number typed were transferred into the mark window and read back. The second word (bits 08, 09, 10, 11) show the read-back data; the direction flop was in the forward direction.

LWN EM

0000

The number shown was decoded as an "END MARK" in error: the correct number is 4100. If the data typed out is 100 00X XXX XXX, the decode gates failed to detect an "EM" condition.

LWN CM

0000

The number shown was decoded as a "CHECK MARK" in error; the correct number is 4240. If the data typed out is 100 01X XXX XXX, the decode gates failed to detect a "CM" condition.

LWN GM

0000

The number shown was decoded as a "GUARD MARK" in error the correct number is 4420. If the data typed out is 100 10X XXX XXX, the decode gates failed to detect a "GM" condition

LWN DM

0000

The number shown was decoded as a "DATA MARK" in error the correct number is 6210. If the data typed out is 110 01X XXX XXX, the decode gates failed to detect a "DM" condition.

LWN FM

0000

The number shown was decoded as a "FINAL MARK" in error the correct number is 6604. If the data typed out is 110 11X XXX XXX, the decode gates failed to detect a "FM" condition.

LWN BM

0000

The number shown was decoded as a "BLOCK MARK" in error the correct number is 7402. If the data typed out is 111 10X XXX XXX, the decode gates failed to detect a "BM" condition.

LWN IM

0000

The number shown was decoded as a "INTERMEDIATE MARK" in error the correct number is 7601. If the data typed out is 111 11X XXX XXX, the decode gates failed to detect a "IM" condition

LWN LC01

0000

The LC01 (1) failed to inhibit decoding of the END MARK, CHECK MARK, GUARD MARK.

LGP GP=GPC INR 11

0000 0000

The data in Bit 11 of the first number typed was transferred to the GP=GPC flop; the data in Bit 11 of the second number is the data actually in the flop.

LGP GP=GPC PRESET

The GP=GPC flop was set to a one, tape preset attempted to clear it, and failed to do so.

LGP GRP DATA

0000 0000

A data transfer was attempted between the MB 0, 1, 2 to the GP register. The first number indicates the data in the MB; the second number indicates the contents of the GP register.

LGP COUNT FAILED

0000 0000

The group counter count logic failed. The first number is the value, the counter should have; the second number is its actual value.

LGP MTP SETUP FAILED TO CLEAR COUNTER

0000

The Magtape instructions, (which generate MTP setup) failed to 0 the group counter. The number typed indicates the contents of the counter after the attempt to clear it has been made.

LGP GROUP COUNTER M115 C25 PINS

E1 F1 H1 J1 RESPOND TO FALSE INPUT

0000

The counter counted when all input gates were disabled.

LGP COMPARE GATES FAILED

0000 0000

Two numbers were loaded one into the GPC, the second into the GP. These numbers were compared by the compare gates with the resultant going into the GP=GPC flip-flop. If the two numbers typed are not equal, then the compare

gates are detecting an equal condition in error. If the values are equal, then the compare gates are not detecting an equal condition.

LGP COUNTS ON FALSE INPUTS

0000

Illegal conditions to the group count logic caused the counter to count.

LTD TAPE FAIL DELAY FAILED TP0

The simulated time pulse TP0 failed to fire off the tape fail delay.

LTD TAPE FAIL MAINT 0 IN PROG 1

The tape fail signal failed to set the motion flop when it timed out the Maintenance flip-flop was set to 0 the in progress flop was set to 1.

LTD TAPE FAIL MAINT 1 IN PROG 1

The tape fail signal set the motion flop in error.
(The tape fail should of been inhibited because the Maintenance flop was set).

LTD TAPE FAIL MAINT 0 IN PROG 0

The tape fail signal set the motion flop in error.
(The tape fail should of been inhibited because the In Progress flop was 0)

LTD ACIP (DIRECTION (1) FAILED)

Setting the direction flop to a (1) failed to fire the ACIP delay.

LTD ACIP (DIRECTION (0) FAILED)

Setting the direction flop to a (0) failed to fire the ACIP delay.

LTD ACIP (MOTION (1) FAILED)

Setting the motion flop to a (1) failed to fire the ACIP delay.

LTD DIRECTION FLOP (SET FORWARD)

"Set Forward" command failed to set the direction flop to (1)

LMU DIRECTION FLOP (SET REVERSE)

"Set Reverse" command failed to set the direction flop to (0)

LMU DIRECTION FLOP (SET REVERSE) MOTION 0 DLY 3

The motion delay 3 failed to set the direction flop to a (0)

LMU DIRECTION FLOP (SET REVERSE) M117 C23

The afore mentioned gate under test failed to set the direction flop to zero.

LMU DIRECTION FLOP M117 C23 FALSE CONDITION

The afore mentioned gate responded to a false input. This could be the result of an open wire or an open input to the gate.

LMU. DIRECTION FLOP M115, C27 MTB

The command "Move Toward Block" (MTB) failed to set the direction flop to the forward direction.

LMU DIRECTION FLOP M115, C27 TAC 10 (1)

The conditions of Block Mark and Search (BM* Search) with TAC 10 (1) and not "MTB" failed to set the direction flop.

LMU DIRECTION FLOP M117 B33 FAILED

The gate under test responded to false conditions.

THE LISTING FOLLOWS

```

/PDP-12 REGISTER TEST (MODEL C PART 1)
/COPYRIGHT 1970, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
/THIS ENTIRE PROGRAM OPERATING IN BOTH PDP-8I AND
/LINC MODE IS DESIGNED TO EXTENSIVELY TEST ALL
/INTERREGISTER TRANSFERS, LINC PROCESSOR ADDERS
/SR00=0 HALT ON ERROR
/SR01=0 PRINT ERROR MESSAGE
/SR02=1 SCOPE MODE ON DATA CAUSING ERROR
/
/SR03=1 SCOPE MODE ON NONERROR TEST
/SR04=0 RING BELL AT END OF TEST

```

```

6151 LMR=6151 /LOAD LINC TAPE MAINTENANCE REGISTER
6152 TRC=6152 /PERFORM OPERATION INDICATED BY PDP-8 ACCUMULATOR
6154 XFR=6154 /PERFORM TRANSFER INDICATED BY MAINTENANCE REGISTER
0003 TAQ=0003 /TRANSFER TAPE ACCUMULATOR TO LINC ACCUMULATOR
0002 PDP=0002 /CHANGE TO PDP-8 PROGRAMMING MODE
6141 LINC=6141 /TRANSFER TO LINC MODE
0001 AOX=0001 /AQ TO EXTEND OPERATIONS BUFFER
0021 XOA=0021
0023 TMA=0023 /AG TO TMA SETUP
0014 NOPL=0014 /LINC MODE NOP
0467 SKPL=0467 /SKIP
0011 CLR=0011
0001 *1

```

```

0001 4100 LOAD, LOADS
0002 3000 K3000, 3000
0003 7000 K7000, 7000
0004 0000 REGA, 0 /OPERATING REGISTER A
0005 0000 REGB, 0 /OPERATING REGISTER B
0006 0000 REGC, 0 /OPERATING REGISTER C
0007 0000 REGD, 0 /OPERATING REGISTER D
0010 0000 REGE, 0 /OPERATING REGISTER E
0011 0000 REGF, 0
0012 0000 PINT, 0
0013 0017 K0017, 0017
0014 0700 K0700, 0700
0015 5400 K5400, 5400
0016 1400 K1400, 1400
0017 0013 K0013, 0013
0020 5170 JMP GQ /START AT 20 START UP
0021 4621 ERROR, ERROR
0022 5000 MAINT1, MAINTS
0023 5052 TYPE, TYPOT
0024 0000 TPEPRE, 0
0025 7330 CLA CLL CML RAR /SET AC00
0026 6152 TRC /GENERATE TAPE PRESET
0027 7200 CLA /CLEAR AC
0030 5424 JMP I TPEPRE /EXIT

```

0031	7400	K7400,	7400
0032	5022	RAN,	RANDM
0033	0000	LSTERR,	0
0034	0000	SPACE,	0
0035	2000	K2000,	2000
0036	4400	K4400,	4400
0037	5000	K5000,	5000
0040	0240	K240,	240
0041	1026	K1026,	1026
0042	0077	K0077,	0077
0043	0010	K0010,	0010
0044	7760	K7760,	7760
0045	6000	K6000,	6000
0046	0004	K0004,	0004
0047	3400	K3400,	3400
0050	0212	K0212,	0212
0051	0020	K0020,	0020
0052	0040	K0040,	0040
0053	0100	K0100,	0100
0054	0600	K0600,	0600
0055	0215	K0215,	0215
0056	0177	K0177,	0177
0057	7776	K7776,	7776
0060	7761	K7761,	7761
0061	7773	K7773,	7773
0062	1000	K1000,	1000
0063	0400	K0400,	0400
0064	5065	LMRWB,	LMRWBS
0065	0003	K0003,	0003
0066	0640	L0FK,	0640
0067	0137	K0137A,	0137
0070	0007	K0007,	0007
0071	7030	K7030,	7030
0072	0070	K0070,	0070
0073	0777	K0777,	0777
0074	7777	K7777A,	7777
0075	0030	K0030,	0030
0076	0050	K0050,	0050
0077	0010	KX0010,	0010
0100	4720	DATUM,	DATUM
0101	4627	ASCRA,	ASCRXT
0102	7455	KTACA,	KTAC

0000	EXIT=0000		
7777	EXITA=7777		
0103 4600	NERROR, NERRDS		
0104 0000	TEMPB, 0		
0105 0207	K0207, 0207		
0106 7740	M40, -40		
0107 0000	GETWIN, 0000		/GET WINDOW SUBROUTINE
0110 7330	CLÄ CLL CML RAR		/SET AC00
0111 6151	LMR		/LOAD MAINT REG
0112 7200	CLÄ		/CLEAR AC
0113 6154	XFR		/READ WINDOW
0114 5507	JMP I GETWIN		/EXIT
0115 7777	K7777, 7777		
0116 0001	K0001, 0001		
0117 0002	K0002, 0002		
0120 0137	K0137, 0137		
0121 0016	K0016, 0016		
0122 4440	K4440, 4440		
0123 7774	K7774, 7774		
0124 6040	K6040, 6040		
0125 7767	K7767, 7767		
0126 7765	K7765, 7765		
0127 7762	K7762, 7762		
0130 5011	BELL, BELLS		
0131 0006	K0006, 0006		
0132 5132	TIMER, TIME		
0133 5252	K5252, 5252		
0134 0152	PNTA, LOCA		
0135 2230	PNTB, LOCB		
0136 5060	PNTC, LOCC		
0137 7405	PNTD, LOCD		
0140 2175	PNTE, KST+1		
0141 2400	K2400, 2400		
0142 7356	K7356, 7356		
0143 1007	K1007, 1007		
0144 2007	K2007, 2007		
0145 5200	TIMTF, TFSUB		
0146 5040	K5040, 5040		
0147 5244	GPSET, GPSETS		
0150 4000	K4000, 4000		
0151 7773	C7773, 7773		
0152 6141	LOCA, LINC		/GO LINC MODE
0153 0700	0700		/COMMAND
0154 0000	0000		/SET GP 00
0155 0002	PDP		/B MODE
0156 5540	JMP I PNTE		
0157 0000	LOADR, 0000		/RETURN ADDRESS STORAGE
0160 3107	DCÄ GETWIN		/STORE DATA
0161 1051	TAD K0020		/GET REVERSE BIT
0162 6152	TRC		/SET REVERSE
0163 7300	CLÄ CLL		/CLEAR AC,L
0164 1107	TAD GETWIN		/GET DATA
0165 4401	JMS I LOAD		/LOAD WINDOW
0166 5557	JMP I LOADR		/EXIT
0167 0200	K0200, 0200		

0170	0040	C0040,	0040
0171	4753	CRLFA,	CRLP
0172	4737	REDOA,	REDO
0173	4762	DATYPA,	DATYP

```

0176      *176
          /
          /RING BELL AT END OF TEST
          /
0176  4024  GO,      JMS TPEPRE      /CLEAR OUT EVERYTHING FOR A NEW START
0177  4530      JMS I BELL        /RETURN FROM END OF PROGRAM
0200  3033      DCA LSTERR       /CLEAR OLD ERROR NUMBER
          /TAPE MAINT REG TC12-0=LTMR
          /DOES MAINTENANCE MODE FLIP FLOP FUNCTION
          /
          /LTMR MAINT MODE FLOP
          /REGB REGC
0201  1004  TMR01,  TAD      REGA      /GET A NEW NUMBER
0202  0052      AND      K0040      /SAVE MAINT MODE BIT
0203  3005      DCA      REGB      /STORE FOR TYPING
0204  1005      TAD      REGB      /FETCH IT
0205  6141      LINC      /GO TO LINC MODE
0206  0001      ADX      /TRANSFER AC TO EXT OP
0207  0011      CLR      /CLEAR AC,L
0210  0021      XOA      /TRANSFER EXT OP TO AC
0211  0002      POP      /GO TO S MODE
0212  0052      AND      K0040      /SAVE ONLY MAINT MODE FLIP FLOP BIT
0213  3006      DCA      REGC      /STORE FOR TYPING
0214  1006      TAD      REGC      /FETCH IT
0215  7041      CIA      /INVERT AND ADD ONE
0216  1005      TAD      REGB      /SUBTRACT DATA SOURCE
0217  7050      SNA CLA  /TEST FOR EQUALITY
0220  4503      JMS I    NERROR     /TEST OKAY
0221  4421      JMS I    ERROR     /TEST FAILED
0222  5476      MAINTM   /MESSAGE TAG
0223  7402      HLT      /ERROR HALT
0224  7610      SKP CLA  /GO TO MONITOR
0225  0201      TMR01   /SCOPE LOOP

```

/DOES TAPE PRESET CLEAR LTMR MAINT FLIP FLOP

0226	1004	TMR02, TAD	REGA	/GET A NEW NUMBER.
0227	0032	AND	K0040	/SAVE MAINT BITS,
0230	3005	DCA	REGB	/STORE FOR TYPING
0231	1005	TAD	REGB	/FETCH IT
0232	6141	LINC		/GO TO LINC MODE
0233	0001	AOX		/TRANSFER AC TO EXT OP REG
0234	0002	PDP		/GO TO 8 MODE
0235	4024	JMS	TPEPRE	/GENERATE TAPE PRESET
0236	6141	LINC		/GO TO LINC MODE
0237	0021	XOA		/TRANSFER EXT OP TO AC
0240	0002	PDP		/GO TO 8 MODE
0241	0032	AND	K0040	/SAVE ONLY THE MAINT MODE FLIP FLOP BIT
0242	3006	DCA	REGC	/STORE FOR TYPING
0243	1006	TAD	REGC	/FETCH IT
0244	7650	SNA CLA		/TEST FOR ZERO
0245	4503	JMS I	NERROR	/TEST OKAY
0246	4421	JMS I	ERROR	/TEST FAILED
0247	5515	MAINTP		/MESSAGE TAG
0250	7402	HLT		/ERROR HALT
0251	7410	SKP		/GO TO MONITOR
0252	0226	TMR02		/SCOPE LOOP

/DOES TAPE PRESET CLEAR LTMR MAINT IR?

0253	1004	TMR03, TAD	REGA	/FETCH A NEW NUMBER
0254	0031	AND	K7400	/SAVE ONLY LTMR MAINT IR BITS0-3
0255	3005	DCA	REGB	/STORE FOR TYPING
0256	1005	TAD	REGB	/FETCH FOR LOADING
0257	6151	LMR		/LOAD MAINTENANCE REGISTER
0260	4024	JMS	TPEPRE	/GENERATE LIP TAPE PRESET
0261	1004	TAD	REGA	/GET A NUMBER
0262	6154	XFR		/"AC TO TB" IF LIP TAPE PRESET OKAY
0263	7300	CLA CLL		/CLEAR
0264	1002	TAD	K3000	/TB TO AC
0265	6151	LMR		/LOAD MAINTENANCE REGISTER
0266	7200	CLA		/CLEAR
0267	6154	XFR		/TB IS TRANSFER TO AC
0270	3006	DCA	REGC	/STORE FOR TYPING
0271	1006	TAD	REGC	/FETCH IT
0272	7041	CIA		/INVERT IT ADD ONE
0273	1004	TAD	REGA	/SUBTRACT DATA SOURCE
0274	7650	SNA CLA		/TEST FOR EQUALITY
0275	4503	JMS I	NERROR	/TEST OKAY
0276	4421	JMS I	ERROR	/TEST FAILED
0277	5422	MAINTC		/MESSAGE TAG
0300	7402	HLT		/ERROR HALT
0301	7410	SKP		/BACK TO MONITOR
0302	0253	TMR03		/SCOPE LOOP

/TEST LTRM IOT 6152 AND TAPE PRESET M112 A33 PINS H2,J2,K2

```

/
TMR04, TAD      K0040      /GET MAINTENANCE FLOP BIT
        LINC          /GO TO LINC MODE
        AOX          /SET MAINTENANCE FLOP
        PDP          /GO TO 8 MODE
        CLA CLL CML RAR /CLEAR LINK SET AC=4000
        TRC      +400      /GENERATE TRC
        TRC      -2        /TEST IOP2
        TRC      +400      /TEST IOT 6152
        TRC      -2        /TEST IOP2
        TRC      +400      /TEST IOT
        TRC      -2        /TEST IOP2
        CLA CLL          /CLEAR
        TRC          /GENERATE 6152
        LINC          /GO TO LINC MODE
        XOA          /TRANSFER EXT OP TO AC
        PDP          /GO TO 8 MODE
        AND      K0040      /SAVE MAINT.
        DCA      REGB      /STORE FOR TYPING
        TAD      REGB      /FETCH IT
        CIA          /INVERT IT ADD ONE
        TAD      K0040      /SUBTRACT MAINT FLOP BIT
        SNA CLA          /TEST FOR EQUALITY
        JMS I  NERROR      /TEST OKAY
        JMS I  ERROR       /TEST FAILED
        MAINT2          /MESSAGE TAG
        HLT          /ERROR HALT
        SKP CLA          /GO TO MONITOR
        TMR04          /SCOPE LOOP
    
```

/DOES IOT 6154 FUNCTION M112, A33 PINS H1,J1,K1 M111 A35 PINS 92,T2

```

/
TMR05, JMS      TPREP      /CLEAR MAINTENANCE REGISTER
        TAD      K5252      /GET 5252
        XFR          /LOAD TB TO 5252
        CLA CMA          /SET AC=7777
        XFR      +400      /TEST IOP4
        XFR      -4        /TEST IOT
        XFR      +400      /TEST IOP4
        XFR      -4        /TEST IOT
        XFR      +400      /TEST IOP4
        XFR      -4        /TEST IOT
        CLA CLL          /CLEAR
        TAD      K3000      /GET "TB TO AC"
        LMR          /LOAD MAINTENANCE REGISTER
        CLA CLL          /CLEAR
        XFR          /TRANSFER TB TO AC(5252)
        DCA      REGB      /STORE FOR TYPING
        TAD      REGB      /FETCH
        CIA          /INVERT AND ADD ONE
        TAD      K5252      /SUBTRACT 5252
        SNA CLA          /TEST FOR EQUALITY
        JMS I  NERROR      /TEST OKAY
        JMS I  ERROR       /TEST FAILED
    
```


0365 5462
0366 7402
0367 7410
0370 0337

MAINT4
HLT
SKP
TMR05

/MESSAGE TAG
/ERROR HALT
/GO TO MONITOR
/SCOPE LOOP

/DOES IOT 6151 FUNCTION M112 A33 PINS L1,M1,N1

```

/THR06,  JMS      TPEPRE      /CLEAR MAINTENANCE REGISTER
          TAD      REGA        /GET DATA
          XFR      /SEND TO TB
          CLA  CLL      /CLEAR AC,L
          TAD      K3000      /GET ALL TB TO AC
          LMR      /SET MAINTENANCE REGISTER TB TO AC
          CLA  CLL      /CLEAR AC,L
          LMR      +400      /DETECT M003
          LMR      +200      /          M004
          LMR      -100      /          M005
          LMR      -40       /          M006
          LMR      +20       /          M007
          LMR      -10       /          M008
          POP      /RETURN TO 8 MODE
          LMR      -1        /DESELECT IOP1
          XFR      /READ TB
          DCA      REGB      /STORE TB
          TAD      REGB      /GET TB
          CIA      /NEGATE
          TAD      REGA      /SUBTRACT DATA SOURCE
          SNA  CLA      /TEST
          JMS  I  NERROR    /TEST OKAY
          JMS  I  ERROR     /TEST FAILED
          MAINTI /MESSAGE TAG
          HLT      /ERROR HALT
          SKP  CLA      /EXIT
          TMR06      /SCOPE LOOP
    
```

/CAN WE SET ALL FLOPS IN THE TAPE MAINT REG

```

/THR07,  CLA  CLL      /CLEAR AC,L
          TAD      K7400      /GET ALL BITS SET
          LMR      /LOAD MAINT REG
          CLA  CLL      /CLEAR AC,L
          XFR      /READ BACK
          DCA      REGB      /STORE FOR TYPING
          TAD      REGB      /FETCH
          SNA  CLA      /TEST FOR = 0000 (UNUSED)
          JMS  I  NERROR    /TEST OKAY
          JMS  I  ERROR     /TEST FAILED
          MAINTX /MESSAGE TAG
          HLT      /ERROR HALT
          SKP  CLA      /EXIT
          TMR07      /SCOPE LOOP
    
```

/THE NEXT 19 ROUTINES TEST AND EXERCISE THE
/LOGIC ON TC12=0=LTRA,B,C,D,E,F

/BINARY SEQUENCE TEST (TB<>AC)

```

0442 6151 TBAC1, LMR /LOAD MAINTENANCE REGISTER TO AC TO TB 00
0443 1004 TAD REGA /GET DATA
0444 3005 DCA REGB /STORE FOR TYPING
0445 1005 TAD REGB /FETCH IT
0446 6154 XFR /TRANSFER IT
0447 7300 CLA CLL /CLEAR ACTIVE REGISTERS
0450 1002 TAD K3000 /GET "TB TO AC" COMMAND
0451 6151 LMR /LOAD MAINTENANCE REGISTER
0452 7300 CLA CLL /CLEAR THE ACCUMULATOR
0453 6154 XFR /MOVE TB TO AC
0454 3006 DCA REGC /STORE FOR TYPING
0455 1006 TAD REGC /GET FOR TESTING
0456 7041 CIA /INVERT DATA
0457 1005 TAD REGB /SUBTRACT SENT DATA
0460 7600 SNA CLA /TEST AND CLEAR
0461 4503 JMS I NERROR /NO PROBLEM CHECK WITH OPERATOR
0462 4421 JMS I ERROR /NOT EQUAL
0463 5066 TBAC /MESSAGE IDENTIFIER
0464 7002 HLT /ERROR HALT
0465 7010 SKP CLA /EXIT AFTER AN ERROR HALT
0466 0442 TBAC1 /SCOPE LOOP ADDRESS
    
```

/RANDOM NUMBER DATA TEST (TB<>AC)

```

0467 6151 TBAC2, LMR /LOAD MAINTENANCE REGISTER "AC TO TB" (00)
0470 4432 JMS I RAN /FETCH A RANDOM NUMBER
0471 3005 DCA REGB /STORE FOR TYPING
0472 1005 TAD REGB /FETCH IT
0473 6154 XFR /"AC TO TB"
0474 7300 CLA CLL /CLEAR AC,L
0475 1150 TAD K4000 /GET IR00=1
0476 6151 LMR /LOAD MAINT REG
0477 6154 XFR /TRY TO LOAD
0500 7300 CLA CLL /CLEAR AC,L
0501 1035 TAD K2000 /GET IR01=1
0502 6151 LMR /LOAD MAINT REG
0503 6154 XFR /TRY TO LOAD
0504 7300 CLA CLL /CLEAR
0505 1002 TAD K1000 /GET MAIN IR02
0506 6151 LMR /LOAD MAINT REG
0507 6154 XFR /TRY TO LOAD
0510 7200 CLA /CLEAR AC
0511 1063 TAD K0400 /GET MAINT IR03
0512 6151 LMR /LOAD MAINT REG
0513 6154 XFR /TRY TO LOAD
0514 7200 CLA /CLEAR AC
0515 1002 TAD K3000 /GET "TB TO AC" (30)
0516 6151 LMR /LOAD MAINTENANCE REGISTER
0517 7300 CLA CLL /CLEAR
    
```

0520 6154
0521 3006
0522 1006
0523 7041
0524 1005
0525 7650
0526 4503
0527 4421
0530 5006
0531 7402
0532 7610
0533 0467

XPR
DCA REGC
TAD REGC
CIA
TAD REGB
SNA CLA
JMS I NERROR
JMS I ERROR
TBAC
HLT
SKP CLA
TBAC2

/WTS TO ACN
/STORE FOR TYPING
/FETCH IT
/INVERT IT
/SUBTRACT SEND DATA
/TEST AND CLEAR
/TEST OKAY
/TEST FAILED
/MESSAGE TAG
/ERROR HALT
/RETURN TO MONITOR
/SCOPE LOOP

/BINARY SEQUENCE TEST (TBN<>AC)

0534	1063	TBNAC1, TAD K0400	/LOAD MAINTENANCE REGISTER TO 00
0535	6151	LMR	/LOAD MAINTENANCE REGISTER
0536	7300	CLÄ CLL	/CLEAR ACCUMULATOR
0537	1004	TAD REGA	/LOAD AC WITH CURRENT AREG
0540	3005	DCA REGB	/STORE FOR TYPING
0541	1005	TAD REGB	/FETCH IT
0542	6154	XFR	/TRANSFER TO TBN
0543	7300	CLÄ CLL	/CLEAR PDP-8 REGISTERS
0544	1141	TAD K2400	/LOAD MAINTENANCE REGISTER TO 05
0545	6151	LMR	/LOAD MAINTENANCE REGISTER
0546	7200	CLA	/CLEAR ACCUMULATOR
0547	6154	XFR	/TRANSFER FROM TBN TO AC
0550	3006	DCA REGC	/STORE FOR TYPING
0551	1006	TAD REGC	/GET FOR TESTING
0552	7041	CIA	/CONVERT TO NEGATIVE NUMBER
0553	1005	TAD REGB	/SUBTRACT SOURCE
0554	7650	SNA CLA	/COMPARE AND CLEAR
0555	4503	JMS I NERROR	/NO PROBLEM
0556	4421	JMS I ERROR	/TROUBLE
0557	5643	TBNAC	/MESSAGE ID
0560	7402	HLT	/ERROR HALT
0561	7610	SKP CLA	/EXIT
0562	0534	TBNAC1	/SCOPE LOOP ADDRESS

/RANDOM NUMBER TEST (TBN<>AC)

0563	1063	TBNAC2, TAD K0400	/GET "AC TO TBN" (01)
0564	6151	LMR	/LOAD MAINT REGISTER
0565	4432	JMS I RAN	/FETCH A RANDOM NUMBER
0566	3005	DCA REGB	/STORE FOR TYPING
0567	1005	TAD REGB	/FETCH IT
0570	6154	XFR	/TRANSFER
0571	7300	CLÄ CLL	/CLEAR
0572	6151	LMR	/LOAD MAINT REG 00
0573	6154	XFR	/TRY TO LOAD TBN
0574	7300	CLÄ CLL	/CLEAR AC:L
0575	1062	TAD K1000	/SET MAIN IR 02
0576	6151	LMR	/LOAD MAINT REG
0577	6154	XFR	/TRY TO LOAD TBN
0600	7300	CLÄ CLL	/CLEAR AC:L
0601	1141	TAD K2400	/GET "TBN TO AC" (21)
0602	6151	LMR	/LOAD MAINT REGISTER
0603	7300	CLÄ CLL	/CLEAR
0604	6154	XFR	/"TBN TO AC"
0605	3006	DCA REGC	/STORE FOR TYPING
0606	1006	TAD REGC	/FETCH IT
0607	7041	CIA	/CONVERT
0610	1005	TAD REGB	/SUBTRACT DATA SOURCE
0611	7650	SNA CLA	/TEST AND CLEAR
0612	4503	JMS I NERROR	/TEST OKAY

0613 4421
0614 5648
0615 7402
0616 7610
0617 0563

JMS I ERROR
TBWAC
HLT
SKP CLA
TBNAC2

/TEST FAILED
/MESSAGE TAG
/ERROR HALT
/RETURN TO MONITOR
/SCOPE LOOP

/BINARY SEQUENCE TEST (TAC<>AC)
/

0620	4024	TACAC1, JMS TPEPRE	/0 IN PROGRESS
0621	1062	TAD K1000	/LOAD MAINTENANCE REGISTER TO 02
0622	6151	LHR	/LOAD MAINTENANCE REGISTER
0623	7200	CLA	/CLEAR OUT OLD COMMAND
0624	1004	TAD REGA	/LOAD AC WITH CURRENT REGA
0625	3005	DCA REGB	/STORE FOR TYPING
0626	1005	TAD REGB	/FETCH IT
0627	6154	XFR	/TRANSFER TO TAC
0630	7200	CLA	/CLEAR ACCUMULATOR
0631	6141	LINC	/GO TO LINC MODE
0632	0003	TAC	/READ TAC
0633	0002	PDP	/PDP-8 MODE
0634	3006	DCA REGC	/STORE FOR TYPING
0635	1006	TAD REGC	/GET IT
0636	7041	CIA	/CONVERT TO NEGATIVE NUMBER
0637	1005	TAD REGB	/ADD SOURCE
0640	7650	SNA CLA	/COMPARE AND CLEAR
0641	4503	JMS I NERROR	/NO ERROR CHECK WITH OPERATOR
0642	4421	JMS I ERROR	/ERROR DATA TRANSFER FAILED
0643	5707	TACAC	/MESSAGE IDENTIFIER
0644	7402	HLT	/ERROR HALT
0645	7610	SKP CLA	/ERROR HALT EXIT
0646	0623	TACAC1*3	/SCOPE LOOP STARTING ADDRESS

/RANDOM NUMBER TEST (TAC<>AC)

```

TACAC2, JMS TPEPRE /0 IN PROGRESS
DCA REGC /0 > REGC
TAD K1000 /GET TAC TO TACN
LHR /LOAD MAINTENANCE REGISTER
JMS I RAN /GET A RANDOM NUMBER
DCA REGB /STORE FOR TYPING
TAD REGB /GET IT
XFR /TAC TO TACN
CLA CLL /CLEAR AC:LL
LHR /LOAD MAINT REG 00
XFR /TRY TO LOAD TAC
CLA CLL /CLEAR AC:LL
TAD K0400 /SET MAIN IR00
LHR /LOAD MAINT REG
XFR /TRY TO LOAD TAC
CLA CLL /CLEAR AC:LL
JMS I MAINT1 /SET NO PAUSE TRY TO READ TAC
LINC /L MODE
0700 /SET IN PROGRESS
0000 /WASTED MEMORY
TAC /LIP IN PROGRESS (0), NOT
PDP /P MODE
DCA REGC /STORE
JMS TPEPRE /IN PROGRESS (0)
LINC /L MODE
0001 /N EQ 03, NOT
PDP /P MODE
TAD REGC /ADD REGC
DCA REGC /STORE IT
LINC /L MODE
0003 /IR07 (0), NOT
PDP /P MODE
TAD REGC /ADD REGC
DCA REGC /STORE
LINC /L MODE
0303 /MBC, NOT
PDP /P MODE
TAD REGC /ADD REGC
DCA REGC /STORE
LINC /GO TO LINC MODE
TAD /TAC TO ACN
PDP /GO TO 8 MODE
TAD REGC /ADD REGC
DCA REGC /STORE FOR TYPING
TAD REGC /FETCH IT
CIA /CONVERT TO 2'S COMP
TAD REGB /SUBTRACT SOURCE DATA
SNA CLA /TEST AND CLEAR
JMS I NERROR /TEST OKAY
JMS I ERROR /TEST FAILED
    
```


0731	9707	TACAC	/MESSAGE TAC
0732	7402	HLT	/ERROR HALT
0733	7610	SKP CLA	/EXIT
0734	0647	TACAC2	/SCOPE LOOP

/BINARY SEQUENCE TEST (TMA SETUP <DAG>)

0735	1004	TASAC1, TAD REGA	/GET DATA
0736	3005	DCA REGB	/STORE FOR TYPING
0737	1005	TAD REGB	/FETCH IT
0740	6144	LINC	/GO TO LINC MODE
0741	0028	TMA	/TRANSFER TO TAPE MA
0742	0008	POP	/POP-B MODE
0743	7008	CLA GLL GML RTR	/LOAD MAINTENANCE REGISTER TO 02
0744	6191	LHR	/LOAD MAINTENANCE REGISTER
0745	7200	CLA	/CLEAR ACCUMULATOR
0746	6154	XFR	/MOVE TMA SETUP TO AC
0747	3006	DCA REGC	/STORE FOR TYPING
0750	1006	TAD REGC	/GET FOR TESTING
0751	7041	CIA	/CONVERT TO REGISTER NUMBER
0752	1009	TAB REGB	/ADD REGB
0753	7000	SNA CLA	/COMPARE AND CLEAR
0754	4000	JMS I NERROR	/NO ERROR CHECK WITH OPERATOR
0755	4421	JMS I ERROR	/ERROR DATA TRANSFER FAILED
0756	0006	TMASET	/MESSAGE IDENTIFIER
0757	7402	HLT	/ERROR HALT
0760	7610	SKP CLA	/EXIT TO NEXT TEST FROM ERROR HALT
0761	0700	TASAC1	/SCOPE LOOP STARTING ADDRESS

/RANDOM NUMBER TEST (TMA SETUP <DAG>)

0762	4400	TASAC2, JMS I RAN	/GET A RANDOM NUMBER
0763	3000	DCA REGB	/STORE FOR TYPING
0764	1000	TAD REGB	/FETCH IT
0765	6144	LINC	/GO TO LINC MODE
0766	0028	TMA	/LOAD TMA SETUP
0767	0011	CLA	/CLEAR AC,L
0770	0407	SRPL	/MBC, NOT
0771	0010	NOPL	/IN CASE OF SKIP
0772	0001	XOA	/EQ 03, NOT
0773	0003	TAC	/IR07(1), NOT
0774	0002	POP	/GO TO B MODE
0775	7002	CLA GLL GML RTR	/GET "TMA SETUP TO AC"
0776	6191	LHR	/LOAD MAINTENANCE REGISTER
0777	7200	CLA	/CLEAR
1000	6194	XFR	/"TMA SETUP TO AC"
1001	3006	DCA REGC	/STORE FOR TYPING
1002	1006	TAD REGC	/FETCH IT
1003	7041	CIA	/CONVERT TO 2'S COMPLEMENT
1004	1009	TAD REGB	/SUBTRACT DATA SOURCE
1005	7000	SNA CLA	/TEST AND CLEAR
1006	4503	JMS I NERROR	/TEST OKAY
1007	4421	JMS I ERROR	/TEST FAILED
1010	6026	TMASET	/MESSAGE TAG
1011	7402	HLT	/ERROR HALT
1012	7610	SKP CLA	/EXIT
1013	0762	TASAC2	/SCOPE LOOP

/BINARY SEQUENCE TEST (TMA<>AC)

1014	1016	TMAAC1, TAD K1400	/GET "AC TO TMA COMMAND
1015	6151	LMR	/LOAD MAINTENANCE REGISTER
1016	7200	CLA	/CLEAR
1017	1004	TAD REGA	/GET DATA SOURCE REGISTER
1020	3005	DCA REGB	/STORE FOR TYPING
1021	1005	TAD REGB	/FETCH IT
1022	6154	XFR	/TRANSFER AC TO TMA
1023	7200	CLA	/CLEAR
1024	1003	TAD K7000	/GET "TMA TO AC COMMAND
1025	6151	LMR	/LOAD MAINTENANCE REGISTER
1026	7200	CLA	/CLEAR
1027	6154	XFR	/TRANSFER TMA TO AC
1030	3006	DCA REGC	/STORE FOR TYPING
1031	1006	TAD REGC	/GET DATA
1032	7041	CIA	/CONVERT TO 2'S COMPLEMENT
1033	1005	TAD REGB	/SUBTRACT DATA SOURCE REGISTER
1034	7650	SNA CLA	/TEST AND CLEAR
1035	4503	JMS I NERROR	/NO ERROR CHECK WITH OPERATOR
1036	4421	JMS I ERROR	/ERROR
1037	5377	TMAC	/MESSAGE IDENTIFIER
1040	7402	HLT	/ERROR HALT
1041	7610	SKP CLA	/EXIT AFTER AN ERROR HALT
1042	1014	TMAAC1	/SCOPE MODE

/RANDOM NUMBER TEST (TMA<>AC)

1043	1016	TMAAC2, TAD K1400	/GET "AC TO TMA"
1044	6151	LMR	/LOAD MAINTENANCE REGISTER
1045	4432	JMS I RAN	/FETCH A RANDOM NUMBER
1046	3005	DCA REGB	/STORE FOR TYPING
1047	1005	TAD REGB	/FETCH IT
1050	6154	XFR	/"AC TO TMA"
1051	7300	CLA CLL	/CLEAR AC,L
1052	1002	TAD K1000	/SET MAINT IR02
1053	6151	LMR	/LOAD MAINT IR
1054	6154	XFR	/TRY TO LOAD TMA
1055	7300	CLA CLL	/CLEAR AC,L
1056	1003	TAD K0400	/SET MAIN IR03
1057	6151	LMR	/LOAD MAINT IR
1060	6154	XFR	/TRY TO LOAD TMA
1061	7300	CLA CLL	/CLEAR AC,L
1062	1003	TAD K7000	/GET "TMA TO AC"
1063	6151	LMR	/LOAD MAINTENANCE REGISTER
1064	7200	CLA	/CLEAR
1065	6154	XFR	/"TMA TO AC"
1066	3006	DCA REGC	/STORE FOR TYPING
1067	1006	TAD REGC	/FETCH IT
1070	7041	CIA	/CONVERT TO 2'S COMP
1071	1005	TAD REGB	/SUBTRACT DATA SOURCE
1072	7650	SNA CLA	/TEST AND CLEAR
1073	4503	JMS I NERROR	/TEST OKAY

1074 4421
1075 9377
1076 7402
1077 7610
1100 1043

JMS 1 ERROR
TMAAC
HLT
SKP CLA
TMAAC2

/TEST FAILED
/MESSAGE TAG
/ERROR MALT
/EXIT
/SCOPE MODE

/BINARY SEQUENCE TEST (AC>TB>RWB>AG)

```

/
TBRWB1, LMR          /LOAD MAINTENANCE REGISTER "A" TO TB"
1101  6151          /GET A TEST NUMBER
1102  1004          /STORE FOR TYPING
1103  3005          /FETCH IT
1104  1005          /"AC TO TB"
1105  6154          /CLEAR
1106  7300          /GET "TB>RWB" COMMAND
1107  1002          /"TB>RWB"
1110  6152          /GET "RWB>AC" COMMAND
1111  1141          /LOAD MAINTENANCE REGISTER
1112  6151          /CLEAR
1113  7300          /"RWB TO AC"
1114  6154          /STORE FOR TYPING
1115  3006          /FETCH IT
1116  1006          /CONVERT TO 2'S COMPLEMENT
1117  7041          /SUBTRACT DATA SOURCE
1120  1005          /TEST FOR EQUALITY
1121  7650          /TEST OKAY
1122  4503          /TEST FAILED
1123  4421          /MESSAGE TAG
1124  6367          /ERROR HALT
1125  7402          /EXIT
1126  7610          /SCOPE LOOP
1127  1101          TBRWB1

```

/RANDOM NUMBER TEST (AC>TB>RWB>AC)

```

/
TBRWB2, LMR          /LOAD MAINTENANCE REGISTER "AC TO TB"
1130  6151          /GET A RANDOM TEST NUMBER
1131  4432          /STORE FOR TYPING
1132  3005          /FETCH IT
1133  1005          /"AC TO TB"
1134  6154          /CLEAR
1135  7300          /GET "TB>RWB" COMMAND
1136  1002          /"TB>RWB"
1137  6152          /GET A RANDOM NUMBER
1140  4432          /LOAD TB WITH A NEW NUMBER
1141  6154          /CLEAR
1142  7300          /GET "RWB>AC" COMMAND
1143  1047          /LOAD MAINTENANCE REGISTER
1144  6151          /CLEAR
1145  7300          /"RWB>AC"
1146  6154          /STORE FOR TYPING
1147  3006          /FETCH IT
1150  1006          /CONVERT TO 2'S COMPLEMENT
1151  7041          /SUBTRACT DATA SOURCE
1152  1005          /TEST AND CLEAR
1153  7650          /TEST OKAY
1154  4503          /TEST FAILED
1155  4421          /MESSAGE TAG
1156  6367          /ERROR HALT
1157  7402          /EXIT
1160  7610          /SCOPE LOOP
1161  1130          TBRWB2

```

/BINARY SEQUENCE TEST (AC>TB>RWB>AC) RWB SHIFT TEST

1162	7300	RWBSH1, CLA CLL	/CLEAR AC,L
1163	6151	LMR	/LOAD MAINTENANCE REGISTER "AC TO TB"
1164	1004	TAD REGA	/GET A TEST NUMBER
1165	3000	DCA REGB	/STORE FOR TYPING
1166	1000	TAD REGB	/FETCH IT
1167	6104	XFR	/"AC TO TB"
1170	7300	CLA CLL	/CLEAR
1171	1060	TAD K1000	/GET "TB>RWB" COMMAND
1172	6100	TRC	/"TB>RWB"
1173	1143	TAD K1007	/SET AC=2007 ("RWB SHIFT")
1174	6100	TRC	/SHIFT RWB ONCE LEFT
1175	1010	TAD K1400	/GET "RWB>AC"
1176	6151	LMR	/LOAD MAINTENANCE REGISTER
1177	7300	CLA CLL	/CLEAR
1200	6154	XFR	/"RWB>AC"
1201	3000	DCA REGC	/STORE FOR TYPING
1202	1000	TAD REGB	/GET SEND DATA
1203	7104	CLL RAL	/SHIFT LEFT
1204	0140	AND K7356	/SAVE SIGNIFICANT DATA
1205	3000	DCA REGB	/SET REGB TO ACTUAL DATA
1206	1000	TAD REGC	/GET RWB DATA FOR TESTING
1207	7041	CIA	/CONVERT TO 2'S COMPLEMENT
1210	1000	TAD REGB	/SUBTRACT SENT DATA
1211	7000	SNA CLA	/TEST FOR EQUALITY
1212	4003	JMS I NERROR	/TEST OKAY
1213	4421	JMS I ERROR	/TEST FAILED
1214	6000	RWBSHF	/MESSAGE TAG
1215	7400	HLT	/ERROR HALT
1216	7010	SKP CLA	/EXIT
1217	1100	RWBSH1	/SCOPE LOOP

/RANDOM NUMBER TEST (A0>TB>RWB>AC) RWB SHIFT TEST

1220	7300	RWBSH2, CLA CLL	/CLEAR AC,L
1221	6151	LMR	/LOAD MAINTENANCE REGISTER "AC TO TB"
1222	4432	JMS I RAN	/GET A RANDOM TEST NUMBER
1223	3005	DCA REGB	/STORE FOR TYPING
1224	1005	TAD REGB	/FETCH IT
1225	6154	XFR	/"AC TO TB"
1226	7300	CLA CLL	/CLEAR
1227	1062	TAD K1000	/GET "TB>RWB" COMMAND
1230	6152	TRC	/"TB>RWB"
1231	4432	JMS I RAN	/GET A RANDOM NUMBER
1232	6154	XFR	/LOAD TB WITH A RANDOM NUMBER
1233	7300	CLA CLL	/CLEAR AC,L
1234	1144	TAD K2007	/GET SHIFT
1235	6152	TRC	/SHIFT RWB ONCE LEFT
1236	1016	TAD K1400	/GET "RWB>AC"
1237	6151	LMR	/LOAD MAINTENANCE REGISTER
1240	7300	CLA CLL	/CLEAR
1241	6154	XFR	/"RWB>AC"
1242	3006	DCA REGC	/STORE FOR TYPING
1243	1005	TAD REGB	/GET SEND DATA
1244	7104	CLL RAL	/SHIFT LEFT
1245	0142	AND K7356	/SAVE SIGNIFICANT DATA
1246	3005	DCA REGB	/SET REGB TO ACTUAL DATA
1247	1006	TAD REGC	/GET RWB DATA FOR TESTING
1250	7041	CIA	/CONVERT TO 2'S COMPLEMENT
1251	1005	TAD REGB	/SUBTRACT SENT DATA
1252	7650	SNA CLA	/TEST FOR EQUALITY
1253	4503	JMS I NERROR	/TEST OKAY
1254	4421	JMS I ERROR	/TEST FAILED
1255	6056	RWBSHF	/MESSAGE TAG
1256	7402	HLT	/ERROR HALT
1257	7510	SKP CLA	/EXIT
1260	1220	RWBSH2	/SCOPE LOOP

/BINARY SEQUENCE TEST MAINT (ACK>RWB) VIA MAINT GATE

1261	1004	RWBMS1, TAD	REGA	/GET A TEST NUMBER
1262	3005	DCA	REGB	/STORE FOR TYPING
1263	1005	TAD	REGB	/FETCH IT
1264	4404	JMS I	LMRWB	/LOAD RWB THRU MAINT GATE
1265	1047	TAD	K3400	/GET "RWB>AC"
1266	6151	LMR		/LOAD MAINTENANCE REGISTER
1267	7300	CLA CLL		/CLEAR
1270	6154	XFR		/"RWB>AC"
1271	3006	DCA	REGC	/STORE FOR TYPING
1272	1006	TAD	REGC	/FETCH IT
1273	7041	CIA		/CONVERT TO 2'S COMPLEMENT
1274	1005	TAD	REGB	/SUBTRACT SEND DATA
1275	7650	SNA CLA		/TEST FOR EQUALITY
1276	4503	JMS I	NERROR	/TEST OKAY
1277	4421	JMS I	ERROR	/TEST FAILED
1280	6334	RWBMG		/MESSAGE TAG
1301	7402	HLT		/ERROR HALT
1302	7610	SKP CLA		/EXIT
1303	1261	RWBMS1		/SCOPE LOOP

/RANDOM NUMBER TEST MAINT (ACK>RWB) VIA MAINT GATE

1304	4422	RWBMS2, JMS I	RAN	/GET A RANDOM TEST NUMBER
1305	3005	DCA	REGB	/STORE FOR TYPING
1306	1005	TAD	REGB	/FETCH IT
1307	4404	JMS I	LMRWB	/LOAD RWB THRU MAINT GATE
1310	1047	TAD	K3400	/GET "RWB>AC"
1311	6151	LMR		/LOAD MAINTENANCE REGISTER
1312	7300	CLA CLL		/CLEAR
1313	6154	XFR		/"RWB>AC"
1314	3006	DCA	REGC	/STORE FOR TYPING
1315	1006	TAD	REGC	/FETCH IT
1316	7041	CIA		/CONVERT TO 2'S COMPLEMENT
1317	1005	TAD	REGB	/SUBTRACT SEND DATA
1320	7650	SNA CLA		/TEST FOR EQUALITY
1321	4503	JMS I	NERROR	/TEST OKAY
1322	4421	JMS I	ERROR	/TEST FAILED
1323	6334	RWBMG		/MESSAGE TAG
1324	7402	HLT		/ERROR HALT
1325	7610	SKP CLA		/EXIT
1326	1304	RWBMS2		/SCOPE LOOP

/BINARY SEQUENCE TEST MB>TBN (EXTENDED ADDRESS)

1327	4024	TBNMB1,	JMS	TPEPRE	/CLEAR EXTENDED ADDRESS BIT
1330	4422		JMS I	MAINT1	/SET NO PAUSE
1331	1004		TAD	REGA	/FETCH A TEST NUMBER
1332	0073		AND	K0777	/SAVE SIGNIFICANT DATA
1333	3005		DCA	REGB	/STORE EXPECTED RESULTS
1334	1004		TAD	REGA	/FETCH FULL 12 BIT TEST NUMBER
1335	3340		DCA	.+3	/STORE IN TEST ADDRESS
1336	6141		LINC		/L MODE
1337	0700		0700		/TAPE COMMAND
1340	0000		0000		/TEST ADDRESS
1341	0002		PDP		/P MODE
1342	1141		TAD	K2400	/GET TBN>AC
1343	6151		LMR		/LOAD MAINT REG
1344	7300		CLA	CLL	/CLEAR AC, LINK
1345	6154		XFR		/READ DATA
1346	3006		DCA	REGC	/STORE RESULTS
1347	1006		TAD	REGC	/FETCH DATA
1350	7041		CIA		/NEGATE
1351	1005		TAD	REGB	/COMPARE
1352	7650		SNA	CLA	/TEST
1353	4503		JMS I	NERROR	/TEST OKAY
1354	4421		JMS I	ERROR	/TEST FAILED
1355	7261		TBX1		/MESSAGE TAG
1356	7402		HLT		/ERROR HALT
1357	7610		SKP	CLA	/EXIT
1360	1327		TBNMB1		/SCOPE LOOP

/RANDOM SEQUENCE TEST MB >TBN (EXTENDED ADDRESS=0)

1361	4024	TBNMB2,	JMS	TPEPRE	/0 EXTENDED ADDRESS
1362	4428		JMS I	MAINT1	/SET MAINT MODE
1363	4488		JMS I	RAN	/GENERATE A RANDOM NUMBER
1364	8873		AND	K8777	/SAVE EXPECTED RESULTS
1365	3888		DCA	REGB	/STORE
1366	1888		TAD	REGB	/FETCH TEST DATA
1367	3372		DCA	.+3	/STORE IN TEST ADDRESS
1370	6141		LINC		/L MODE
1371	8787		8787		/TAPE COMMAND
1372	8888		8888		/TEST ADDRESS
1373	8882		POP		/P MODE
1374	1141		TAB	K2488	/GET TBN TAG
1375	6181		LMR		/LOAD MAIN REG
1376	7888		CLA	CLL	/CLEAR AC,L
1377	6184		XFR		/READ DATA
1400	3888		DCA	REGC	/STORE RESULTS
1401	1888		TAD	REGC	/FETCH DATA
1402	7841		CIA		/NEGATE
1403	1888		TAD	REGB	/SUBTRACT DATA SOURCE
1404	7888		SNA	CLA	/TEST
1405	4888		JMS I	NERROR	/TEST ORAY
1406	4481		JMS I	ERROR	/TEST FAILED
1407	7281		TBX1		/MESSAGE TAG
1410	7488		HLT		/ERROR HALT
1411	7818		SKP	CLA	/EXIT
1412	1881		TBNMB2		/SCOPE LOOP

/
/BINARY SEQUENCE TEST MB >TBN (EXTENDED ADDRESS=1)

```

/
TBNMB3, TAD      REGA      /FETCH A TEST NUMBER
1413 1004
1414 3005      DCA      REGB      /STORE FOR TYPING
1415 1005      TAD      REGB      /FETCH FOR TESTING
1416 3223      DCA      ,+B      /STORE IN TEST ADDRESS
1417 1072      TAD      K0070     /SET EXT ADD, NO PAUSE, MAINT
1420 6141      LINC                     /L MODE
1421 0001      AOX                     /SET NO PAUSE
1422 0705      0705          /TAPE COMMAND
1423 0000      0000          /TEST ADDRESS
1424 0002      PDP                     /P MODE
1425 7300      CLA CLL                     /CLEAR AC,L
1426 1141      TAD      K2400     /GET TBN>AC
1427 6151      LMR                     /LOAD MAINT REG
1430 7300      CLA CLL                     /CLEAR AC, L
1431 6154      XFR                     /READ DATA
1432 3006      DCA      REGC      /STORE RESULTS
1433 1006      TAD      REGC      /FETCH RESULTS
1434 7041      CIA                     /NEGATE
1435 1005      TAD      REGB      /SUBTRACT DATA SOURCE
1436 7650      SNA CLA                     /TEST
1437 4503      JMS I   NERROR      /TEST OKAY
1440 4421      JMS I   ERROR       /TEST FAILED
1441 7305      TBN2                      /MESSAGE TAG
1442 7402      HLT                     /ERROR HALT
1443 7610      SKP CLA                     /EXIT
1444 1413      TBNMB3                    /SCOPE LOOP

```

/
/RANDOM SEQUENCE TEST MB>T0N (EXTENDED ADDRESS=1)

```

/
TBNMB4, JMS I   RAN      /GENERATE A RANDOM NUMBER
1445 4432
1446 3005      DCA      REGB      /STORE FOR TYPING
1447 1005      TAD      REGB      /FETCH FOR TESTING
1450 3253      DCA      ,+B      /STORE IN TEST ADDRESS
1451 1072      TAD      K0070     /SET EXT ADD, NO PAUSE, MAINT
1452 6141      LINC                     /L MODE
1453 0001      AOX                     /SET NO PAUSE
1454 0702      0702          /TAPE COMMAND
1455 0000      0000          /TEST ADDRESS
1456 0002      PDP                     /P MODE
1457 7300      CLA CLL                     /CLEAR AC,L
1460 1141      TAD      K2400     /GET TBN > AC
1461 6151      LMR                     /LOAD MAINT REG
1462 7300      CLA CLL                     /CLEAR AC,L
1463 6154      XFR                     /READ DATA
1464 3006      DCA      REGC      /STORE RESULTS
1465 1006      TAD      REGC      /FETCH IT
1466 7041      CIA                     /NEGATE
1467 1005      TAD      REGB      /SUBTRACT DATA SOURCE
1470 7650      SNA CLA                     /TEST
1471 4503      JMS I   NERROR      /TEST OKAY

```

1472 4421
1473 7305
1474 7402
1475 5301
1476 1445

JMS I ERROR
TBX2
HLT
JMP ADDT81
TBNMB4

/TEST FAILED
/MESSAGE TAG
/ERROR HALT
/EXIT
/SCOPE LOOP

/LTR: TAPE REGISTER ADDER TEST 1 FLOAT 0 THRU TB FLOAT 1 THRU TAC

1477	7300	CLL	CLA		
1500	5304	JMP		,+4	
1501	4024	ADTS1,	JMS	TPEPRE	/0 > IN PROGRESS
1502	7360	CMA	CLL	CML CLA	/SET AC=7777 L=1
1503	3005	DCA		REGB	/STORE IT
1504	1005	TAD		REGB	/FETCH IT
1505	7030	CML	RAR		/SHIFT RIGHT
1506	7420	SNL			/ZERO FLOATED OUT YET
1507	5302	JMP		ADTS1+1	/FLOATING ZERO SUNK START OVER
1510	3005	DCA		REGB	/STORE RESULTANT
1511	7120	CLL	CML		/SET AC=0000 L=1
1512	3006	DCA		REGC	/STORE IT
1513	1006	TAD		REGC	/FETCH IT
1514	7010	RAR			/FLOAT A ONE
1515	3006	DCA		REGC	/STORE RESULTANT
1516	7630	SRL	CLA		/ONE FLOATED OUT YET
1517	5277	JMP		ADTS1+2	/START OVER
1520	1005	TAD		REGB	/SIMULATE THE LTR ADDITION
1521	1006	TAD		REGC	/+REGC
1522	3007	DCA		REGC	/STORE SIMULATION
1523	6151	LMR			/START LTR ADDITION SETUP
1524	1005	TAD		REGB	/FETCH REGB
1525	6154	XFR			/REGB TO TB
1526	7300	CLA	CLL		/CLEAR
1527	1062	TAD		K1000	/GET "AC TO TAC" COMMAND
1530	6151	LMR			/LOAD MAINTENANCE REGISTER
1531	7300	CLA	CLL		/CLEAR
1532	1006	TAD		REGC	/FETCH REGC
1533	6154	XFR			/REGC TO TAC
1534	7300	CLA	CLL		/CLEAR
1535	1063	TAD		K0400	/GET TB & TAC TO TAC
1536	6152	TRC			/DO IT
1537	7300	CLA	CLL		/CLEAR
1540	6141	LINC			/LINC MODE
1541	0003	TAC			/READ TAC
1542	0002	PDP			/B MODE
1543	3010	DCA		REGE	/STORE IT
1544	1010	TAD		REGE	/FETCH IT
1545	7041	CIA			/NEGATE
1546	1007	TAD		REGD	/SUBTRACT SIMULATION
1547	7650	SNA	CLA		/TEST AND CLEAR
1550	4503	JMS	I	NERROR	/TEST OKAY
1551	4421	JMS	I	ERROR	/TEST FAILED
1552	5731	TBADD			/MESSAGE TAG
1553	7402	HLT			/ERROR HALT
1554	5360	JMP		ADTS2	/EXIT
1555	1513			ADTS1+12	

```

/ LTR: TAPE REGISTER ADDER TEST 2 FLOAT B THRU TAC FLOAT 1 THRU TB
1555 7300 CLA CLL /CLEAR L, AC
1557 5362 JMP ,+3 /CONTINUE

ADDTS2, CMÄ CML CLA CLL /SET AC=7777 L=1
1560 7360 DCA REGC /STORE IT
1561 3006 TAD REGC /FETCH IT
1562 1006 CML RAR /INVERT IT SHIFT RIGHT
1563 7030 SNL /DONE YET
1564 7420 JMP ADDTS2 /REDO
1565 5360 DCA REGC /STORE RESULTANT
1566 3006 CLL CML /AC=0000 L=1
1567 7120 DCA REGB /STORE IT
1570 3005 TAD REGB /FETCH IT
1571 1005 RAR /SHIFT RIGHT
1572 7010 DCA REGB /STORE RESULTANT
1573 3005 SEL CLA /DONE YET
1574 7430 JMP ADDTS2=2 /START OVER
1575 5306 TAD REGB /SIMULATE LTR ARCITION
1576 1005 TAD REGC /*REGC
1577 1006 DCA REGD /STORE SIMULATION
1600 3007 LMR /SET MAINTENANCE TO 00 AC TO TB
1601 6101 TAD REGB /GET REGB
1602 1005 XFR /REGB TO TB
1603 6104 CLA CLL /CLEAR
1604 7300 TAD K1000 /GET "AC TO TAC" COMMAND
1605 1002 LMR /LOAD MAINTENANCE REGISTER
1606 6101 CLA CLL /CLEAR
1607 7300 TAD REGC /FETCH REGC
1610 1006 XFR /REGC TO TAC
1611 6104 CLA CLL /CLEAR
1612 7300 TAD K0400 /GET TB +TAC TO TAC
1613 1003 TRQ /DO IT
1614 6102 CLA CLL /CLEAR
1615 7300 LINC /LINC MODE
1616 6141 TAC /GET "TAC"
1617 0003 PDP /8 MODE
1620 0002 DCA REGE /STORE IT
1621 3010 TAD REGE /GET IT
1622 1010 CIA /NEGATE
1623 7041 TAD REGD /SUBTRACT SIMULATION
1624 1007 SNA CLA /TEST AND CLEAN
1625 7650 JMS I NERROR /TEST OKAY
1626 4503 JMS I ERROR /TEST FAILED
1627 4421 TBADD /MESSAGE TAG
1630 5731 HLT /ERROR HALT
1631 7402 SKP CLA /EXIT
1632 7610 ADDTS2+11 /SCOPE LOOP
1633 1571

```

/TAPE R
/
/TAPE REGISTER ADDER EXERCISER
/

1634	4432	EXAD1, JMS I RAN	/GENERATE A RANDOM NUMBER
1635	3005	DCA REGB	/LOAD REGB
1636	4432	JMS I RAN	/GENERATE A RANDOM NUMBER
1637	3006	DCA REGC	/LOAD REGC
1640	6151	LMR	/SET MAINTENANCE REGISTER TO "AC TO TB"
1641	1005	TAD REGB	/GET TB SOURCE REGISTER
1642	6154	XFR	/TRANSFER REGB TO TB
1643	7200	CLA	/CLEAR DATA
1644	1062	TAD K1000	/GET "AC TO TAC" COMMAND
1645	6151	LMR	/LOAD MAINTENANCE REGISTER
1646	7300	CLA CLL	/CLEAR PREVIOUS INSTRUCTION
1647	1006	TAD REGC	/GET TAC SOURCE REGISTER
1650	6154	XFR	/TRANSFER REGA TO TAC
1651	7200	CLA	/CLEAR DATA
1652	1063	TAD K0400	/GET "TB+TAC TO TAC" COMMAND
1653	6152	TRC	/EXECUTE IT
1654	7200	CLA	/CLEAR PREVIOUS COMMAND
1655	1005	TAD REGB	/START OF SIMULATION
1656	1006	TAD REGC	/GET TERMS ADDED
1657	3007	DCA REGD	/STORE FOR FUTURE REFERENCE AND PRINTING
1660	6141	LINC	/GO TO LINC
1661	0003	TAC	/GET TAC
1662	0002	PDP	/GO TO PDP-8
1663	3010	DCA REGE	/STORE FOR FUTURE REFERENCE AND PRINTING
1664	1010	TAD REGE	/GET TRUE SUM
1665	7041	CIA	/CONVERT TO A NEGATIVE NUMBER
1666	1007	TAD REGD	/ADD SIMULATION SUM
1667	7550	SNA CLA	/COMPARE AND CLEAR
1670	4503	JMS I NERROR	/OK CHECK WITH OPERATOR
1671	4421	JMS I ERROR	/ERROR
1672	3731	TSADD	/MESSAGE IDENTIFIER
1673	7402	HLT	/ERROR HALT
1674	7610	SKP CLA	/EXIT HALT ERROR HALT
1675	1634	EXAD1	/SCOPE MODE STARTING ADDRESS

/TC12-B-LCX TESTS TAPE EXTENDED OPERATIONS

/DOES TAPE EXT OP REG LOAD AND READ ALL NUMBERS

1675	7604	LAS		/GET KEYS
1677	0167	AND	K0200	/SAVE TAPE FIELDS BIT
1700	7650	SNA	CLA	/IS IT SET
1701	1003	TAD	K7000	/NO ADD TAPE FIELD TEST
1702	1120	TAD	K0137	/ADD CONSTANT
1703	3007	DCA	K0137A	/MODIFY
1704	7604	LAS		/GET KEYS
1705	0167	AND	K0200	/SAVE TRAP FIELDS BIT
1706	7650	SNA	CLA	/IS IT SET
1707	1003	TAD	K7000	/NO ADD FIELD BITS
1710	1073	TAD	K0777	/FETCH CONSTANT
1711	3074	DCA	K7777A	/MODIFY MASK
1712	1004	LCTX01, TAD	REGA	/GET A NEW NUMBER
1713	0067	AND	K0137A	/SAVE ALL BUT MARK AND MAINT MODE
1714	1052	TAD	K0040	/SET MAINT FLOP BIT
1715	3005	DCA	REGB	/STORE FOR TYPING
1716	1005	TAD	REGB	/FETCH IT
1717	1054	TAD	K0600	/TRY TO SET MARK AND UNUSED BIT
1720	6141	LINC		/LINC MODE
1721	0001	AOX		/LOAD EXT OP REG
1722	0011	CLR		/CLEAR AC AND LINK
1723	0021	XOA		/READ BACK "EXT OP REG"
1724	0002	PDP		/GO TO 8 MODE
1725	0074	AND	K7777A	/SAVE SIGNIF
1726	3006	DCA	REGC	/STORE RXED DATA FOR TYPING
1727	1006	TAD	REGC	/GET IT
1730	7041	CIA		/INVERT ADD ONE
1731	1005	TAD	REGB	/SUBTRACT SEND DATA
1732	7600	SNA	CLA	/TEST FOR EQUALITY
1733	4003	JMS	I NERROR	/TEST OKAY
1734	4421	JMS	I ERROR	/TEST FAILED
1735	6256	LCXDT1		/MESSAGE TAG
1736	7402	HLT		/ERROR HALT
1737	7610	SKP	CLA	/RETURN TO MONITOR
1740	1712	LCX01		/SCOPE LOOP

/DOES TAPE PRESET CLEAR THE EXT OP REG

1741	1067	LCX02, TAD	K0137A	/GET ALL ONES
1742	1040	TAD	K240	/SET MAINT, MARK FLOP
1743	6141	LINC		/GO TO LINC MODE
1744	0001	AOX		/LOAD EXT OP REG
1745	0002	PDP		/GO TO 8 MODE
1746	4024	JMS	TPEPRE	/GENERATE PRESET
1747	6141	LINC		/GO TO LINC MODE
1750	0021	XOA		/READ EXT OP REG
1751	0002	PDP		/GO TO 8 MODE
1752	0073	AND	K0777	/SAVE
1753	3005	DCA	REGB	/STORE FOR TYPING
1754	1005	TAD	REGB	/FETCH IT
1755	7650	SNA	CLA	/TEST FOR ALL ZEROS

1756	4503	JMS I	NERROR	/TEST OKAY
1757	4421	JMS I	ERROR	/TEST FAILED
1760	6277	LCXTPS		/MESSAGE TAG
1761	7402	HLT		/ERROR HALT
1762	7610	SKP CLA		/RETURN TO MONITOR
1763	1741	LCX02		/SCOPE LOOP

/DOES "LOAD EXT OP REG GATE M117 B33 S1 FUNCTION

1754	1067	LCX03, TAD	K0137A	/GOT ALL ONES
1755	1040	TAD	K240	/ADD MAINT MODE AND MARK FLOP
1766	6141	LINC		/GOT TO LINC MODE
1767	0001	AOX		/LOAD EXT OP REG
1770	0011	CLR		/CLEAR ALL
1771	0016	NOPL		/TEST WITH N EQ 01 (NOT)
1772	0301	0301		/TEST WITH MSC (NOT)
1773	0021	X0A		/READ BACK "EXT OP REG"
1774	0002	POP		/GO TO 0 MODE
1775	0074	AND	K7777A	/SAVE
1776	3005	DCA	REGB	/STORE FOR TYPING
1777	1005	TAD	REGB	/FETCH IT
2000	7041	CIA		/INVERT ADD ONE
2001	1067	TAD	K0137A	/SUBTRACT TEST DATA
2002	1052	TAD	K0040	/SUBTRACT MAINT MODE
2003	7050	SNA CLA		/TEST FOR EQUALITY
2004	4503	JMS I	NERROR	/TEST OKAY
2005	4421	JMS I	ERROR	/TEST FAILED
2006	6316	LCXLTD		/MESSAGE TAG
2007	7402	HLT		/ERROR HALT
2010	7010	SKP CLA		/RETURN TO MONITOR
2011	1704	LCX03		/SCOPE LOOP

/ DOES EXT OP REG WORK WITH HIGH SPEED RANDOM NUMBERS

2012	4432	LCX04,	JMS I	RAN	/GENERATE A RANDOM NUMBER
2013	0067		AND	K0137A	/SAVE ALL BUT MARK AND MAINT
2014	1092		TAD	K0040	/ADD MAINT MODE
2015	3005		DCA	REGB	/STORE FOR TYPING
2016	1005		TAD	REGB	/FETCH IT
2017	1054		TAD	K0000	/TRY TO SET MARK AND UNUSED
2020	6141		LINC		/GO TO LINC MODE
2021	0001		AOX		/LOAD "EXT OP REG"
2022	0021		XOA		/READ "EXT OPREG"
2023	0001		AOX		/ETC
2024	0021		XOA		/ETC
2025	0001		AOX		/ETC
2026	0021		XOA		
2027	0001		AOX		
2030	0021		XOA		
2031	0001		AOX		
2032	0021		XOA		
2033	0002		PDP		/GO TO 0 MODE
2034	0074		AND	K7777A	/SAVE
2035	3006		DCA	REGC	/STORE FOR TYPING
2036	1006		TAD	REGC	/FETCH IT
2037	7041		CIA		/INVERT ADD ONE
2040	1005		TAD	REGB	/SUBTRACT DATA SOURCE
2041	7650		SNA CLA		/TEST FOR EQUALITY
2042	4503		JMS I	NERROR	/TEST OKAY
2043	4421		JMS I	ERROR	/TEST FAILED
2044	6256		LCXDT1		/MESSAGE TAG
2045	7402		HLT		/ERROR HALT
2046	7610		SKP CLA		/EXIT
2047	2012		LCX04		/SCOPE LOOP

/ DOES EX-OP READ BACK M177 C23,V21 M111 C21,E1 FUNCTION

2050	1067	LCX05,	TAD	K0137A	/SET ALL FLOPS
2051	1040		TAD	K240	/SET
2052	6141		LINC		/L MODE
2053	0001		AOX		/SET EXTENDED OPS
2054	0011		CLR		/CLEAR AC, L
2055	0002		PDP		/P MODE
2056	1004		TAD	REGA	/GET A TEST NUMBER
2057	7004		RAL		/JUSTIFY LEFT
2060	6141		LINC		/L MODE
2061	0016		NOPL		/TEST WITH N E0 01 (NOT)
2062	0321		0321		/TEST WITH MSC (NOT)
2063	0700		0700		/TEST WITH PROGRESS = 0 (NOT)
2064	7000		7000		/WASTED MEMORY
2065	0002		PDP		/P MODE
2066	3005		DCA	REGB	/STORE AC
2067	1005		TAD	REGB	/FETCH OLD AC
2070	7041		CIA		/NEGATE
2071	1004		TAD	REGA	/SUBTRACT DATA SOURCE
2072	7650		SNA CLA		/TEST RESULTS

2073	4503	JMS I	NERROR	/TEST OKAY
2074	4421	JMS I	ERROR	/TEST FAILED
2075	6437	LCXRB1		/MESSAGE TAG
2076	7402	HLT		/ERROR HALT
2077	4502	JMS I	KTACA	/EXIT
2100	2050	LCX05		/SCOPE LOOP

```

/TEST LCX LDF 03, 04
/
LDFT0: JMS      TPREP
        JMS I   MAINT1
        TAD     K0020
        TRC
        CLA CLL
        TAD     K7774
        DCA     K7773
        TAD     REGA
        AND     K0003
        DCA     REGC
        TAD     REGC
        TAD     LDFK
        DCA     ,+2
        LINC
        0000
        0700
        7000
        PDP
        TAD     K0040
        LMR
        CLA CLL
        TAD     K0016
        JMS I   LOAD
        TAD     K7030
        LMR
        CLA CLL
        XFR
        CLL RTL
        RAJ
        AND     K0003
        DCA     REGD
        TAD     REGD
        CIA
        TAD     REGC
        SNA CLA
        JMS I   NERROR
        JMS I   ERROR
        TAG
        HLT
        SKP CLA
LDFT0
/SET EVERYTHING TO ZERO
/SET MAINT MODE
/SET DIRECTIONAL FLOP=0
/SET
/CLEAR AC,L
/GET NEW COUNT
/UPDATE LOAD COUNT
/FETCH A TEST WORD
/SAVE DF 03, 04
/STORE FOR TYPING
/GET DATA FIELD
/ADD LDF COMMAND
/STORE IT
/L MODE
/CHANGE DATA FIELD (LCX TOP 03,04)
/SET TINR 11 (0)
/SET LOP GP 00 (1)
/P MODE
/GET IT0
/GENERATE TP0,1
/CLEAR AC, L
/GET BLOCK MARK
/LOAD WINDOW
/GET TMA > AC AND GENERATE TP3,TP4
/LOAD MAINTENANCE REGISTER
/CLEAR AC,L
/READ DATA
/SHIFT TB0,1 INTO AC10,11
/SHIFT
/SAVE AC10,11
/STORE FOR TYPING
/FETCH FOR TESTING
/NEGATE
/SUBTRACT SOURCE DATA
/TEST RESULTS
/TEST OKAY
/TEST FAILED
/MESSAGE TAG
/ERROR HALT
/EXIT
/SCOPE LOOP

```

```

/TEST LCX LIF 03 04
LIFT0, JMS TPEPRE /SET EVERYTHING TO ZERO
        JMS I MAINT1 /SET MAINT MODE
        TAD K0020 /SET DIRECTION (0)
        TRC /SET
        CLA GLL /CLEAR AC, L
        TAD K7774 /GET NEW COUNT
        DCA K7773 /UPDATE LOAD COUNT
        TAD REGA /FETCH A TEST WORD
        AND K0003 /SAVE IF 03,04
        DCA REGC /STORE FOR TYPING
        TAD REGC /GET DATA FIELD
        DCA DISPAT /ASS BASIC JUMP
DISPAT, 0000 /DISPATCH
        JMP I PNTA /GO TO FIELD 0
        JMP I PNTB /GO TO FIELD 1
        JMP I PNTC /GO TO FIELD 2
        JMP I PNTD /GO TO FIELD 3
KST, JMP I DISPAT+1 /CONSTANT
        CLA GLL /CLEAR AC,L
        TAD K0040 /SET TTS
        LMR /LOAD MAINTENANCE REGISTER
        CLA GLL /CLEAR AC, L
        TAD K0016 /GET BLOCK MARK
        JMS I LOAD /LOAD WINDOW
        TAD K7030 /GET IMA > AC
        LMR /LOAD MAINTENANCE REGISTER
        CLA GLL /CLEAR AC,L
        XFR /READ DATA
        GLL RTL /MOVE AC0,1 TO AC10,11
        RAL /SHIFT LEFT
        AND K0003 /SAVE SIGNIFICANT DATA
        DCA REGD /STORE FOR TYPING
        TAD REGD /FETCH FOR TESTING
        CIA /NEGATE
        TAD REGC /SUBTRACT DATA SOURCE
        SNA CLA /TEST
        JMS I NERROR /TEST OKAY
        JMS I ERROR /TEST FAILED
        TAGA /MESSAGE TAG
        HLT /ERROR HALT
        SKP CLA /EXIT
        LIFT0 /SCOPE LOOP

/RESET THE CONSTANT
2225 1151 TAD C7773 /SET AC=7773
2226 3061 DCA K7773 /RESTORE COUNT
2227 5235 JMP LIN01 /JUMP OVER INSTRUCTION FIELD ROUTINE

```

/ INSTRUCTION FIELD ROUTINE

2230	6141	LOCB,	LINC		/L MODE
2231	0700		0700		/TAPE COMMAND
2232	0000		0000		/WASTE
2233	0002		PDP		/P MODE
2234	5540		JMP I	PNTE	/EXIT

/TC12-0=LIN
/NEXT 7 TESTS

/DOES TAC=7777 GATE FUNCTION I.E. DETECT TAC=7777

2235	1002	LIN01,	TAD	K1000	/GET "AC TO TAC" COMMAND
2236	6151		LMR		/LOAD MAINTENANCE REGISTER
2237	7240		CLA	CMA	/SET AC=7777
2240	6154		XPR		/TRANSFER 7777 TO TAC
2241	7300		CLA	CLL	/CLEAR
2242	6141		LINC		/GO TO LINC MODE
2243	0003		TAC		/TAPE AC TO CP AC
2244	0002		PDP		/GO TO 0 MODE
2245	3005		DCA	REGB	/STORE TAC FOR TYPING
2246	1036		TAD	K4400	/GET "STATES & TO AC" COMMAND
2247	6151		LMR		/LOAD MAINTENANCE REGISTER
2250	7300		CLA	CLL	/CLEAR
2251	6154		XPR		/READ STATUS WORD
2252	7710		SPA	CLA	/TEST TAC=7777
2253	4503		JMS I	NERROR	/TEST OKAY
2254	4421		JMS I	ERROR	/FAILED
2255	5553		LIN77		/MESSAGE TAG
2256	7402		HLT		/ERROR HALT
2257	7410		SKP		/RETURN TO MONITOR
2260	2235		LIN01		/SCOPE LOOP

/DOES TAG=7777 GATE DETECT NON 7777 CONDITIONS

2251	7360	LIN02	CLA	CLL	CMA	CML	/SET AC=7777 L=1
2252	3005		DCA		REGB		/STORE FOR TYPING
2263	1005		TAD		REGB		/FETCH IT
2264	7030		RAR	CML			/SHIFT RIGHT COMPLEMENT LINK
2265	7420		SNL				/IS THE ZERO IN THE LINK
2266	5261		JMP		LIN02		/START OVER
2267	3005		DCA		REGB		/STORE FOR TYPING
2270	1062		TAD		K1000		/GET "AC TO TAG" COMMAND
2271	6151		LMR				/LOAD MAINTENANCE REGISTER
2272	7300		CLA	CLL			/CLEAR
2273	1005		TAD		REGB		/GET THE TEST NUMBER
2274	6154		XFR				/TRANSFER NUMBER TO "TAG"
2275	7300		CLA	CLL			/CLEAR
2276	1036		TAD		K4400		/GET "STATES & TIMING TO AC"
2277	6151		LMR				/LOAD MAINTENANCE REGISTER
2300	7300		CLA	CLL			/CLEAR
2301	6154		XFR				/TRANSFER STATES & TIMING TO AC
2302	7700		SMA	CLA			/TEST AC00
2303	4903		JMS	I	NERROR		/TEST OKAY
2304	4421		JMS	I	ERROR		/TEST FAILED
2305	5553		LIN77				/MESSAGE TAG
2306	7402		HLT				/ERROR HALT
2307	7410		SKP				/GO BACK TO MONITOR
2310	2263		LIN02*				/SCOPE LOOP

/ DOES LIP TAPE PRESET CLEAR LIN TMR

2311	4422	LIN04,	JMS I	MAINT1	/SET MAINT MODE
2312	1004		TAD	REGA	/GET A TEST NUMBER
2313	0070		AND	K0007	/SAVE SIGNIFICANT DIGITS
2314	3005		DCA	REGB	/STORE FOR TYPING
2315	1005		TAD	REGB	/FETCH IT
2316	1014		TAD	K0700	/ADD BASIC COMMAND
2317	3321		DCA	,+2	/SET UP PSEUDO INSTRUCTION
2320	6141		LINC		/GO TO LINC MODE
2321	0000		0000		/TEST INSTRUCTION
2322	3000		3000		/WASTED SPACE
2323	0002		PDP		/GO TO 8 MODE
2324	4024		JMS	TPEPRE	/GENERATE TAPE PRESET
2325	1037		TAD	K3000	/GET "INSTRUCTION TO AC" COMMAND
2326	1063		TAD	K0400	/ADD BIT 03
2327	6151		LHR		/LOAD MAINTENANCE REGISTER
2330	7300		CLA CLL		/CLEAR
2331	6154		XFR		/TRANSFER "INSTRUCTION TO AC"
2332	0044		AND	K7760	/SAVE SIGNIFICANT DATA
2333	3006		DCA	REGC	/STORE FOR TYPING
2334	7330		CLA CLL	CML RAR	/SET RDC
2335	1006		TAD	REGC	/FETCH
2336	7650		SNA CLA		/TEST FOR EQUALITY
2337	4503		JMS I	NERROR	/TEST OKAY
2340	4421		JMS I	ERROR	/TEST FAILED
2341	5571		LINIPS		/MESSAGE TAG
2342	7402		HLT		/ERROR HALT
2343	7410		SKP		/GO TO MONITOR
2344	2311		LIN04		/SCOPE LOOP

/DOES LIN INSTRUCTION DECODER FUNCTION (BINARY SEQUENCE)

2345	4422	LIN05,	JMS I	MAINT1	/SET MAINT MODE
2346	1004		TAD	REGA	/GET A TEST NUMBER
2347	8070		AND	K0007	/SAVE SIGNIFICANT DIGITS
2350	3000		DCA	RE00	/STORE FOR TYPING
2351	1000		TAD	RE00	/FETCH IT
2352	1014		TAD	K0700	/ADD BASIC INSTRUCTION
2353	5555		DCA	+2	/SET UP PSEUDO INSTRUCTION
2354	6141		LINC		/GO TO LINC MODE
2355	8000		0000		/TEST INSTRUCTION
2356	3000		3000		/WASTED SPACE
2357	0002		POP		/GO TO 0 MODE
2360	7500		CLA	CLL	/CLEAR
2361	1007		TAD	K5000	/GET "INSTRUCTION TO AC" COMMAND
2362	1005		TAD	K0400	/ADD BIT 05
2363	6101		LHR		/LOAD MAINTENANCE REGISTER
2364	7000		CLA	CLL	/CLEAR
2365	6104		XFR		/TRANSFER INSTRUCTION TO AC
2366	8044		AND	K7760	/SAVE SIGNIFICANT DATA
2367	3006		DCA	RE0C	/STORE FOR TYPING
2370	1000		TAD	RE0B	/GET INSTRUCTION
2371	7040		CMA		/INVERT
2372	3011		DCA	REGF	/STORE TALLY NUMBER
2373	7000		CLL	CLA	/SET LINK
2374	7010		RAR		/SHIFT RIGHT
2375	2011		ISE	REGF	/DONE YET
2376	5374		JMP	+2	/NOPE
2377	7041		CIA		/YEP INVERT ADD ONE
2400	1006		TAD	REGC	/SUBTRACT RECEIVED DATA
2401	7050		SNA	CLA	/TEST FOR EQUALITY
2402	4500		JMS I	NERROR	/TEST OKAY
2403	4421		JMS I	ERROR	/TEST FAILED
2404	5610		LININD		/MESSAGE TAG
2405	7402		HLT		/ERROR HALT
2406	7410		SKP		/GO BACK TO MONITOR
2407	2345		LIN05		/SCOPE LOOP

/DOES LIN INSTRUCTION DECODER FUNCTION (RANDOM NUMBERS)

2410	4422	LIN06	JMS I	MAINT1	/SET NO PAUSE
2411	4432		JMS I	RAN	/GET A RANDOM NUMBER
2412	0070		AND	K0007	/SAVE SIGNIFICANT DIGITS
2413	3005		DCA	REGB	/STORE FOR TYPING
2414	1005		TAD	REGB	/FETCH IT
2415	1014		TAD	K0700	/ADD BASIC INSTRUCTION
2416	3220		DCA	,+2	/SET UP PSEUDO INSTRUCTION
2417	6141		LINC		/L MODE
2420	0000		0000		/TEST INSTRUCTION
2421	3000		3000		/WASTED MEMORY
2422	0002		PDP		/B MODE
2423	7300		CLA	CLL	/CLEAR AC,L
2424	1037		TAD	K5000	/GET INST TO AC
2425	1063		TAD	K0400	/ADD IR03(1)
2426	6151		LMR		/LOAD MAINTENANCE REGISTER
2427	7300		CLA	CLL	/CLEAR AC,L
2430	6154		XFR		/TRANSFER TO AC
2431	0044		AND	K7760	/SAVE SIGNIFICANT DATA
2432	3006		DCA	REGC	/STORE FOR TYPING
2433	1005		TAD	REGB	/GET INSTRUCTION
2434	7040		CMÄ		/INVERT IT
2435	3011		DCA	REGF	/STORE TALLY NUMBER
2436	7320		CLA	CLL CML	/SET LINK
2437	7010		RAR		/SHIFT RIGHT
2440	2011		ISE	REGF	/DONE YET?
2441	5237		JMP	,+2	/NOPE
2442	7041		CIA		/YEP! NEGATE
2443	1006		TAD	REGC	/SUBTRACT RECEIVED DATA
2444	7050		SNA	CLA	/TEST
2445	4503		JMS I	NERROR	/TEST OKAY
2446	4421		JMS I	ERROR	/TEST FAILED
2447	5610		LININD		/MESSAGE TAG
2450	7402		HLT		/ERROR HALT
2451	7410		SKP		/EXIT
2452	2410		LIN06		/SCOPE LOOP

/DOES LIN I BIT FUNCTION? (RANDOM SEQUENCE)

2453	4422	LIN07,	JMS I	MAINT1	/SET NO PAUSE AND MAINT MODE
2454	4432		JMS I	RAN	/GET A NUMBER
2455	0051		AND	K0020	/SAVE I BIT
2456	3005		DCA	REGB	/STORE FOR TYPING
2457	1005		TAD	REGB	/FETCH IT
2460	1014		TAD	K0700	/ADD BASIC COMMAND
2461	3263		DCA	,+2	/STORE PSEUDO OPERATION
2462	6141		LINC		/GO TO LINC MODE
2463	0000		0000		/PSEUDO COMMAND
2464	3000		3000		/WASTE
2465	0000		POP		/GO TO 0 MODE
2466	1007		TAD	K5000	/GET INST TO AC
2467	1063		TAD	K0400	/ADD LMR BIT 03
2470	0351		LNR		/LOAD MAINTENANCE REGISTER
2471	7000		CLA	CLL	/CLEAR
2472	0154		XPR		/TRANSFER TO AC
2473	0043		AND	K0010	/SAVE I BIT ONLY
2474	7104		CLL	RAL	/SHIFT LEFT
2475	3000		DCA	REGB	/STORE FOR TYPING
2476	1000		TAD	REGB	/FETCH IT
2477	7040		CRA		/INVERT ADD ONE
2500	1000		TAD	REGB	/SUBTRACT SEND DATA
2501	7000		SNA	CLA	/TEST FOR EQUALITY
2502	4500		JMS I	NERROR	/TEST OWAY
2503	4401		JMS I	ERROR	/TEST FAILED
2504	5030		LENIDT		/MESSAGE TAG
2505	7400		HLT		/ERROR HALT
2506	7410		SKP		/GO TO MONITOR
2507	2454		LIN07+1		/SCOPE LOOP

/TC12-0-LTC

/TEST UNIT SELECT DATA LOADING AND DECODE GATES (BINARY SEQUENCE)

2510	1004	UNIDTA, TAD REGA	/GET DATA
2511	3005	DCA REGB	/STORE FOR TYPING
2512	1005	TAD REGB	/FETCH
2513	0131	AND K0006	/SAVE AC9,10
2514	7110	RAR CLL	/SHIFT AC9,10 INTO 10, AND 11
2515	1043	TAD K0010	/ADD NO PAUSE
2516	6141	LINC	/GO TO LINC MODE
2517	0001	AOX	/AC TO EXTEND OPS
2520	0002	PDP	/GO TO 8 MODE
2521	7300	CLA CLL	/CLEAR
2522	1005	TAD REGB	/GET DATA SOURCE
2523	7006	RTL	/MOVE AC11, TO AC9
2524	7004	RAL	/PUT BIT 9 INTO 8
2525	0043	AND K0010	/SAVE
2526	1065	TAD K0003	/ADD MTB
2527	1014	TAD K0700	/ADD BASIC TAPE COMMAND
2530	3332	DCA ,+2	/STORE IT
2531	6141	LINC	/GO TO LINC MODE
2532	0000	0000	/BASIC TAPE COMMAND
2533	7000	7000	/WASTED SPACE
2534	0002	PDP	/GO TO 8 MODE
2535	1037	TAD K5000	/GET UNITS TO AC
2536	6151	LMR	/LOAD MAINTENANCE REGISTER
2537	7200	CLA	/CLEAR
2540	6154	XFR	/TRANSFER
2541	7040	CMA	/INVERT IT
2542	0044	AND K7760	/SAVE ONLY UNIT DECODES
2543	3007	DCA REGD	/STORE FOR TYPING
2544	1005	TAD REGB	/GET
2545	0070	AND K0007	/SAVE AC9, 10 11
2546	7040	CMA	/INVERT DATA
2547	3011	DCA REGF	/STORE IT
2550	7120	CLL CML	/SET LINK
2551	7010	RAR	/SHIFT RIGHT
2552	2011	ISE REGF	/DONE YET
2553	5351	JMP ,+2	/NOPE
2554	3006	DCA REGC	/STORE SIM CODE
2555	1006	TAD REGC	/FETCH SIM CODE
2556	7041	CIA	/NEGATE
2557	1007	TAD REGD	/SUBTRACT
2560	7650	SNA CLA	/TEST
2561	4503	JMS I NERROR	/TEST OK
2562	4421	JMS I ERROR	/TROUBLE
2563	6072	UNIDAT	/ERROR MESSAGE
2564	7402	HLT	/ERROR HALT
2565	7610	SKP CLA	/EXIT
2566	2510	UNIDTA	/SCOPE LOOP

/UNIT SELECT TAPE PRESET TEST

2557	1017	UNTPRE, TAD K0013	/MTB + BIT(00) 1
2558	1052	TAD K0040	/SET MAINT
2559	0141	LINC	/LINC MODE
2572	0001	AOX	/AC TO EXTENDED OPS
2573	0713	0713	/COMMAND
2574	7000	7000	/WASTE SPACE
2575	0002	PDP	/GO TO B MODE
2576	4024	JMS TPEPRE	/GET "UNITS TO AC"
2577	1043	TAD K0010	/LOAD MAINTENANCE REGISTER
2580	1052	TAD K0040	/SET MAINT
2581	0141	LINC	/LINC MODE
2582	0001	AOX	/AC TO EX SET NO PAUSE
2583	0002	PDP	/GO TO B MODE
2584	7300	CLA CLL	/CLEAR AC.L
2585	0194	XFR	/TRANSFER
2586	0044	AND K7760	/SAVE SIGNIFICANT DATA
2587	3005	DCA REG8	/STORE FOR TYPING
2510	1005	TAD REG8	/GET IT
2511	7710	SPA CLA	/TEST AND CLEAR
2512	0226	JMP ,+14	/GOOF
2513	1017	TAD K0013	/SET MTB + BIT (00) 1
2514	0141	LINC	/GO TO LINC MODE
2515	0703	0703	/TAPE INSTRUCTION
2516	3000	3000	/PREUNIT TO UNIT
2517	0002	PDP	/P MODE
2520	0154	XFR	/TRANSFER
2521	0044	AND K7760	/SAVE SIGNIFICANT DUTY
2522	3005	DCA REG8	/STORE FOR TYPING
2523	1005	TAD REG8	/GET IT
2524	7700	SMA CLA	/TEST AND CLEAR
2525	4503	JMS I NERROR	/NO TROUBLE
2526	4421	JMS I ERROR	/TROUBLE!
2527	0110	UNTIMES	/UNIT PRESET MESSAGE
2530	7402	HLT	/ERROR HALT
2531	7410	SKP CLA	/EXIT AFTER ERROR HALT
2532	2507	UNTPRE	/SCOPE LOOP

/SET UNIT 1 TEST

2633	4422	UNIONE, JMS I MAINT1	/SET NO PAUSE
2634	6141	LINC	/GO TO LINC MODE
2635	0703	0703	/SET UNIT ZERO
2636	7000	7000	/WASTE SPACE
2637	0002	PDP	/GO TO 8 MODE
2640	7300	CLA CLL	/CLEAR AC,L
2641	1052	TAD K0040	/SET UNIT 1
2642	6152	TRC	/SET IT
2643	7300	CLA CLL	/CLEAR AC,L
2644	1037	TAD K5000	/GET UNITS TO AC
2645	6151	LMR	/LOAD MAINTENANCE REGISTER
2646	7300	CLA CLL	/CLEAR AC,L
2647	6154	XFR	/TRANSFER
2650	7040	CMA	/INVERT DATA
2651	0044	AND K7760	/SAVE SIGNIFICANT DATA
2652	3005	DCA REG8	/STORE FOR TYPING
2653	1005	TAD REG8	/FETCH FOR TESTING
2654	1150	TAD K4000	/SUBTRACT UNIT 1
2655	7650	SNA CLA	/TEST
2656	4503	JMS I NERROR	/TEST OKAY
2657	4421	JMS I ERROR	/TEST FAILED
2660	6535	UNQNM	/ERROR MESSAGE
2661	7402	HLT	/ERROR HALT
2662	7610	SKP CLA	/EXIT
2663	2633	UNIONE	/SCOPE LOOP

/TEST UNIT SELECT LOGIC FOR (ONE UNIT)

2664	1037	UNSEL, TAD K5000	/GET UNITS TO AC
2665	6151	LMR	/LOAD MAINT REG
2666	7300	CLA CLL	/CLEAR AC,L
2667	6154	XFR	/READ UNITS MOTION
2670	0117	AND K0002	/SAVE UNIT SEL
2671	3005	DCA REG8	/STORE FOR TYPING
2672	1005	TAD REG8	/FETCH IT
2673	7112	CLL RTR	/MOVE AC10 TO LINK
2674	7630	SEL CLA	/TEST OK
2675	4503	JMS I NERROR	/TEST OK
2676	4421	JMS I ERROR	/TEST FAILED
2677	6571	UNSELM	/MESSAGE TAG
2700	7402	HLT	/ERROR HALT
2701	7610	SKP CLA	/EXIT
2702	2664	UNSEL	/SCOPE LOOP

/TEST WRITE ENABLE LOGIC

2703	1037	WRTENB, TAD K5000	/GET UNITS TO AC
2704	6151	LMR	/LOAD MAINT REG
2705	7300	CLA CLL	/CLEAR AC, L
2706	6154	XFR	/READ UNITS MOTION

2707	0116	AND	K0001	/SAVE AC11
2710	3005	DCA	REGB	/STORE FOR TYPING
2711	1005	TAD	REGB	/FETCH IT
2712	7110	CLL	RAR	/MOVE AC11 INTO LINK
2713	7630	SZL	CLA	/TEST ENABLE BIT
2714	4503	JMS	I NERROR	/TEST OK
2715	4421	JMS	I ERROR	/TEST FAILED
2716	6552	WREN		/MESSAGE TAG
2717	7402	HLT		/ERROR HALT
2720	7610	SKP	CLA	/EXIT
2721	2703	WRTEN		/SCOPE LOOP

/LWN DATA TRANSFER DIRECTION(1) FORWARD BINARY SEQUENCE

2720	4024	LWNDR1, JMS TPEPRE	/SET MOTION TO ZERO
2721	1003	TAD K2100	/SET AC05
2722	6152	TRC	/SET DIRECTION TO ZERO
2723	1200	CLL	/CLEAR
2724	1004	TAD RECA	/GET TALLY NUMBER
2727	1010	OCA REGE	/STORE FOR TYPING
2730	1010	TAD REGE	/FETCH
2731	7040	OHA	/INVERT DATA
2735	4421	JMS I LOAD	/LOAD THE MARK WINDOW
2736	1005	TAD REG9	/GET DATA
2739	7012	RTR CLL	/ALIGN WINDOW DATA WITH SENT DATA
2739	7012	RTR	/MOVE IT
2739	7012	RTR	/MOVE IT
2737	7013	RAR	/MOVE IT
2740	1005	OCA REG9	/STORE FOR PRINTING
2741	1005	TAD REG9	/GET IT
2742	7041	CIA	/MAKE IT NEGATIVE NUMBER
2743	1005	TAD REGE	/ADD TALLY
2744	0013	AND K0017	/SAVE SIGNIFICANT DATA
2745	7650	SNA CLA	/COMPARE AND CLEAR
2746	4503	JMS I NERROR	/TEST OKAY
2747	4421	JMS I ERROR	/DATA SENT NOT SAME AS DATA RECEIVED
2750	5766	DATABD	/MESSAGE NAME
2751	7402	HLT	/ERROR HALT
2752	7510	SKP CLA	/EXIT TO NEXT TEST AFTER ERROR HALT
2753	2722	LWNDR1	/SCOPE MODE STARTING ADDRESS

/LWN DATA TRANSFER DIRECTION(0) REVERSE RANDOM SEQUENCE

2754	4024	LWNRN2, JMS TPEPRE	/0>MOTION
2755	1001	TAD K0020	/SET AC07
2756	6152	TRC	/SET DIRECTION TO 1
2757	4432	JMS I RAN	/FETCH A RANDOM NUMBER
2758	1010	OCA REGE	/STORE FOR TYPING
2761	1010	TAD REGE	/FETCH IT
2762	4401	JMS I LOAD	/LOAD THE WINDOW
2763	1005	TAD REG9	/GET DATA
2764	7112	RTR CLL	/ALIGN WINDOW DATA WITH SENT DATA
2765	7012	RTR	/MOVE
2766	7012	RTR	/MOVE
2767	7010	RAR	/MOVE
2770	3025	OCA REG9	/STORE FOR TYPING
2771	1010	TAD REGE	/FETCH IT
2772	7041	CIA	/NEGATE
2773	1005	TAD REG9	/SUBTRACT
2774	0013	AND K0017	/SAVE SIGNIFICANT DATA
2775	7650	SNA CLA	/TEST
2776	4503	JMS I NERROR	/TEST OKAY
2777	4421	JMS I ERROR	/TEST FAILED
3000	6005	DATABD	/MESSAGE TAG
3001	7402	HLT	/ERROR HALT
3002	7510	SKP CLA	/EXIT

/PDP-12 REGISTER TEST (MODEL C PART 1) PAL10

V141

27-AU

2121

PAGE 38-1

0003 2754

LWNRN2

/SCOPE LOOP

/LWN DATA TRANSFER DIRECTION (1) FORWARD RANDOM SEQUENCE

```

LWNRN1, JMS TPEPRE      /0 > MOTION
          TAD K0100      /SET AC05
          TRC            /SET DIRECTION TO ONE
          JMS I RAN      /FETCH A RANDOM NUMBER
          DCA REGE       /STORE FOR TYPING
          TAD REGE       /FETCH
          CMA CLL        /INVERT IT
          JMS I LOAD     /LOAD THE WINDOW
          TAD REGB       /GET DATA
          RTR CLL        /ALIGN WINDOW DATA WITH SENT DATA
          RTR            /MOVE
          RTR            /MOVE
          RAR            /MOVE
          DCA REGB       /STORE IT
          TAD REGB       /FETCH IT
          CIA            /NEGATE
          TAD REGE       /SUBTRACT
          AND K0017      /SAVE SIGNIFICANT DATA
          SNA CLA        /TEST AC AND CLEAR
          JMS I NERROR   /TEST OK
          JMS I ERROR    /TEST FAILED
          DATABD         /ERROR MESSAGE
          HLT            /ERROR HALT
          SKP CLA        /EXIT
          LWNRN1        /SCOPE LOOP
    
```

/LWN DATA TRANSFER DIRECTION(0) REVERSE RANDOM SEQUENCE

```

LWNDR2, JMS TPEPRE      /0 > MOTION
          TAD K0020      /SET DIRECTION TO (0)
          TRC            /SET IT
          CLA            /CLEAR AC
          TAD REGA       /GET THE TALLY NUMBER TO SEND
          DCA REGE       /STORE FOR TYPING
          TAD REGE       /FETCH IT
          JMS I LOAD     /LOAD MARK WINDOW
          TAD REGB       /GET DATA
          RTR CLL        /ALIGN WINDOW DATA WITH SENT DATA
          RTR            /MOVE IT
          RTR            /MOVE
          RAR            /MOVE
          DCA REGB       /STORE FOR PRINT OUT
          TAD REGB       /GET IT FOR COMPARE
          CIA            /ADD ONE
          TAD REGE       /ADD SEND NUMBER
          AND K0017      /SAVE SIGNIFICANT DATA
          SNA CLA        /COMPARE AND CLEAR
          JMS I NERROR   /SO FAR SO GOOD
          JMS I ERROR    /GO TO ERROR PROCESSOR
          DATAFW        /MESSAGE NAME
          HLT            /ERROR HALT
    
```

3064 7610
3065 3035

SKP CLA /EXIT AFTER ERROR HALT
LWNDT2 /SCOPE MODE STARTING ADDRESS

/SUBROUTINE TO SET LTS LC01(1)

3066 1051
3067 6152
3070 7300
3071 1121
3072 4401
3073 1075
3074 6151
3075 7300

TAD K0020 /SET DIRECTION REVERSE
TRC /SET IT
CLA CLL /CLEAR AC,L
TAD K0016 /GET BM
JMS I LOAD /LOAD WINDOW TO BLOCK MARK
TAD K0030 /GET TP3,TP4
LMR /GENERATE TP3,TP4
CLA CLL /CLEAR AC,L

```

/
/ THE NEXT FIVE(5) ROUTINES LOAD ALL COMBINATIONS OF NUMBERS
/ INTO THE MARK WINDOW, THEN IT CHECKS TO SEE IF
/ THE LOADED NUMBER IS THE SELECTED MARK NUMBER
/ IS SO IT ASCERTAINS IF THE SELECTED IS SET
/ IF THE LOADED NUMBER IS NOT SELECTED IT
/ CHECKS TO ASCERTAIN THAT SELECTED IS NOT UP
/

```

3076	1004	EMTST,	TAD REGA	/GET TALLY TO SEND IT
3077	4401		JMS I LOAD	/LOAD MARK WINDOW
3100	1004		TAD REGA	/GET TALLY NUMBER
3101	0013		AND K0017	/SAVE SIGNIFICANT DIGIT
3102	7041		CIA	/INVERT IT
3103	7640		SZA CLA	/EQUAL
3104	7100		CLL	/NO CLEAR LINK
3105	1005		TAD REGB	/GET RXED DATA
3106	0053		AND K0100	/SAVE END MARK
3107	7650		SNA CLA	/WAS IT SET
3110	7020		CML	/NO INVERT LINK
3111	7430		SZL	/LINK ZERO
3112	4503		JMS I NERROR	/TEST OK
3113	4421		JMS I ERROR	/ERROR PROCESSOR
3114	5370		END	/MESSAGE NAME
3115	7402		HLT	/ERROR HALT
3116	7610		SKP CLA	/EXIT AFTER ERROR HALT
3117	3076		EMTST	/SCOPE MODE STARTING ADDRESS

/LWN CM CHECKSUM MARK TEST
/

3120	1004	CMTST,	TAD REGA	/GET TALLY TO SEND IT
3121	4401		JMS I LOAD	/LOAD MARK WINDOW
3122	1004		TAD REGA	/GET TALLY NUMBER
3123	0013		AND K0017	/SAVE WINDOW DATA
3124	1115		TAD K7777	/SUBTRACT ONE
3125	7640		SZA CLA	/EQUAL?
3126	7100		CLL	/NO CLEAR LINK
3127	1005		TAD REGB	/GET RXED DATA
3130	0052		AND K0040	/SAVE CHECK MARK BIT
3131	7650		SNA CLA	/HAS IT SET
3132	7020		CML	/NO INVERT LINK
3133	7430		SZL	/LINK = 0
3134	4503		JMS I NERROR	/TEST OK
3135	4421		JMS I ERROR	/ERROR PORCESSOR
3136	5316		CHECK	/CHECK MARK DECODE DOESN'T
3137	7402		HLT	/ERROR HALT
3140	7610		SKP CLA	/EXIT AFTER ERROR HALT
3141	3120		CMTST	/SCOPE MODE ADDRESS

/LWN GM GUARD MARK TEST
/

3142	1004	GMTST,	TAD REGA	/GET TALLY TO SEND IT
3143	4401		JMS I LOAD	/LOAD MARK WINDOW
3144	1004		TAD REGA	/GET TALLY NUMBER
3145	0013		AND K0017	/SAVE SIGNIFICANT DATA
3146	1057		TAD K7776	/SUBTRACT GUARD MARK
3147	7640		SZA CLA	/EQUAL?
3150	7100		CLL	/NO CLEAR LINK
3151	1005		TAD REGB	/GET RXED DATA
3152	0051		AND K0020	/SAVE GUARD MARK BIT
3153	7650		SNA CLA	/HAS IT SET
3154	7020		CML	/NO INVERT LINK
3155	7430		SZL	/LINK = 0
3156	4503		JMS I NERROR	/TEST OK
3157	4421		JMS I ERROR	/ERROR PROCESSOR
3160	5325		GUARD	/GUARD MARK DECODE DOESN'T
3161	7402		HLT	/ERROR HALT
3162	7610		SKP CLA	/EXIT AFTER ERROR HALT
3163	3142		GMTST	/SCOPE MODE ADDRESS

/LWN DM DATA MARK TEST
/

3164	1004	DMTST,	TAD REGA	/GET TALLY TO SEND IT
3165	4401		JMS I LOAD	/LOAD MARK WINDOW

3166	1004	TAD REGA	/GET TALLY NUMBER
3167	0013	AND K0017	/SAVE SIGNIFICANT DATA
3170	1125	TAD K7767	/SUBTRACT DATA MARK
3171	7640	SEA CLA	/EQUAL?
3172	7100	CLL	/NO CLEAR LINK
3173	1005	TAD REGB	/GET RXED DATA
3174	0043	AND K0010	/SAVE DATA MARK BIT
3175	7650	SNA CLA	/WAS IT SET
3176	7020	CML	/NO INVERT LINK
3177	7430	SZL	/LINK = 0
3200	4503	JMS I NERROR	/TEST OK
3201	4421	JMS I ERROR	/ERROR PROCESSOR
3202	5334	DATA	/DATA MARK DECODE DOESN'T
3203	7402	HLT	/ERROR HALT
3204	7610	SKP CLA	/EXIT AFTER ERROR HALT
3205	3164	DNTST	/SCOPE MODE ADDRESS

/LWN FM FINAL MARK TEST

3206	1004	FMTST, TAD REGA	/GET TALLY TO SEND IT
3207	4401	JMS I LOAD	/LOAD MARK WINDOW
3210	1004	TAD REGA	/GET TALLY NUMBER
3211	0013	AND K0017	/SAVE WINDOW DATA
3212	1126	TAD K7765	/SUBTRACT FINAL MARK
3213	7640	SEA CLA	/EQUAL?
3214	7100	CLL	/NO CLEAR LINK
3215	1005	TAD REGB	/GET RXED DATA
3216	0046	AND K0004	/SAVE FINAL MARK BIT
3217	7650	SNA CLA	/WAS IT SET
3220	7020	CML	/NO INVERT LINK
3221	7430	SZL	/LINK = 0
3222	4503	JMS I NERROR	/TEST OK
3223	4421	JMS I ERROR	/ERROR PROCESSOR
3224	5343	FINAL	/FINAL MARK DECODE DOESN'T
3225	7402	HLT	/ERROR HALT
3226	7610	SKP CLA	/EXIT AFTER ERROR HALT
3227	3206	FMTST	/SCOPE MODE ADDRESS

/LWN BM BLOCK MARK TEST

3230	1004	BMTST, TAD REGA	/GET TALLY TO SEND IT
3231	4401	JMS I LOAD	/LOAD MARK WINDOW
3232	1004	TAD REGA	/GET TALLY NUMBER
3233	0013	AND K0017	/SAVE WINDOW DATA
3234	1127	TAD K7762	/SUBTRACT BLOCK MARK
3235	7640	SEA CLA	/EQUAL?
3236	7100	CLL	/NO CLEAR LINK
3237	1005	TAD REGB	/GET RXED DATA
3240	0117	AND K0002	/SAVE CHECK MARK BIT
3241	7650	SNA CLA	/WAS IT SET
3242	7020	CML	/NO INVERT LINK
3243	7430	SZL	/LINK = 0

/TAPE 3

3244	4503	JMS I NERROR	/TEST OK
3245	4421	JMS I ERROR	/ERROR PROCESSOR
3246	5352	BLOCK	/BLOCK MARK DECODE DOESN'T
3247	7402	HLT	/ERROR HALT
3250	7610	SKP CLA	/EXIT AFTER ERROR HALT
3251	3230	BHTST	/SCOPE MODE ADDRESS

/LWN IM INTERMEDIATE MARK TEST

3252	1004	IMTST.	TAD REGA	/GET TALLY TO SEND IT
3253	4401	JMS I LOAD	/LOAD MARK WINDOW	
3254	1004	TAD REGA	/GET TALLY NUMBER	
3255	0013	AND K0017	/SAVE SIGNIFICANT DATA	
3256	1060	TAD K7761	/SUBTRACT IM MARK	
3257	7640	SEA CLA	/EQUAL?	
3260	7100	CLL	/NO CLEAR LINK	
3261	1005	TAD REGB	/GET RXED DATA	
3262	0116	AND K0001	/SAVE INTERMEDIATE MARK BIT	
3263	7650	SNA CLA	/WAS IT SET	
3264	7020	CML	/NO INVERT LINK	
3265	7430	SEL	/LINK = 0	
3266	4503	JMS I NERROR	/TEST OK	
3267	4421	JMS I ERROR	/ERROR PROCESSOR	
3270	5361	INTERM	/INTERM MARK DECODE DOESN'T	
3271	7402	HLT	/ERROR HALT	
3272	7610	SKP CLA	/EXIT AFTER ERROR HALT	
3273	3252	IMTST	/SCOPE MODE ADDRESS	

/DOES LC01(0) INHIBIT DECODING OF EM, QM, GM

3274	1121	LC010,	TAD K0016	/GET BM
3275	4401	JMS I LOAD	/SET WINDOW TO BM	
3276	1075	TAD K0030	/SET FOR TP3, TP4	
3277	6151	LMR	/GENERATE TP3 SET LC01, LC00	
3300	1043	TAD K0010	/SET AC = 0040 SET FOR TP0, 1, 2	
3301	6151	LMR	/GEN TP0, TP1, TP2 0 LC00	
3302	7110	CLL RAR	/SET AC = 0020 SET FOR TP3, TP4	
3303	6151	LMR	/GENERATE TP3 0>LC01, LC00	
3304	7300	CLA CLL	/CLEAR AC, L	
3305	4401	JMS I LOAD	/SET WINDOW TO EM	
3306	4107	JMS GETWIN	/GET MARK WINDOW	
3307	3005	DCA REGB	/STORE FOR TYPING	
3310	1005	TAD REGB	/FETCH FOR TYPING	
3311	7041	CIA	/NEGATE	
3312	1150	TAD K4000	/SUBTRACT CORRECT DATA	
3313	7650	SNA CLA	/TEST	
3314	4503	JMS I NERROR	/TEST OKAY	
3315	4421	JMS I ERROR	/TEST FAILED	
3316	7331	LC0M	/MESSAGE 100	
3317	7402	HLT	/ERROR HALT	
3320	7610	SKP CLA	/EXIT	
3321	3274	LC010	/SCOPE LOOP	

/TAPE GROUP COUNTER TC12-B-LGP

/GR EQ GPC INR 11 DATA TRANSFER M216,B37,PIN T2,S2,M115,A36,P2,R2,S2

```

3322 4422      GPCNT1, JMS I MAINT1  /SET NO PAUSE
3323 1004      TAD REGA          /GET A NUMBER
3324 0116      AND K0001      /SAVE AC11
3325 3005      DCA REG0      /STORE FOR TYPING
3326 1005      TAD REG0      /FETCH IT
3327 1014      TAD K0700      /ADD BASIC TAPE
3330 3332      DCA ,+2        /STOW IT
3331 6141      LINC          /GO TO LINC MODE
3332 0000      0000          /TEST LOCATION
3333 7000      7000          /SET LGP GP
3334 0002      PDP           /8 MODE
3335 1124      TAD K0040      /GET STATUS=AC
3336 6151      LMR           /LOAD MAINTENANCE REGISTER
3337 7300      CLA CLL        /CLEAR
3340 6154      XFR           /TRANSFER
3341 7012      RTR           /MOVE TOWARD AC11
3342 7050      RAR OMA        /MOVE AND INVERT IT
3343 0116      AND K0001      /SAVE LGP GPCNT=GP
3344 3006      DCA REGC      /STORE FOR TYPING
3345 1006      TAD REGC      /GET IT.
3346 7041      CIA           /INVERT IT.
3347 1005      TAD REG0      /ADD SENDING DATA
3350 7050      SNA CLA        /TEST COMPARISON
3351 4503      JMS I NERROR   /NO PROBLEM
3352 4421      JMS I ERROR    /ERROR
3353 6127      GPE0B         /MESSAGE ID
3354 7402      HLT           /ERROR HALT
3355 7010      SKP CLA        /EXIT AFTER
3356 3322      GPCNT1        /SCOPE LOOP
    
```

/DOES TAPE PRESET CLEAR GP EQ GPC? M216,B37 PIN K2

```

3357 4422      GPCNT2, JMS I MAINT1  /SET NO PAUSE
3360 6141      LINC          /LINC MODE
3361 0701      0701          /SET LGP=GP
3362 7000      7000          /WASTE
3363 0002      PDP           /GO TO 8 MODE
3364 1052      TAD K0040      /GET TP2
3365 6151      LMR           /GENERATE TP2
3366 4024      JMS TPEPRE     /TAPE PRESET
3367 1045      TAD K0000      /STATES TO AC
3370 6151      LMR           /LOAD MAINTENANCE REGISTER
3371 7200      CLA           /CLEAR
3372 6154      XFR           /TRANSFER
3373 0043      AND K0010      /SAVE BIT
3374 7050      SNA CLA        /TEST
3375 4503      JMS I NERROR   /NO PROBLEMS
3376 4421      JMS I ERROR    /TROUBLE
3377 6150      GPPRE         /MESSAGE ID
3400 7402      HLT           /ERROR HALT
    
```

1101 7610
1102 3387

SKP CLA
GPCNT2

/EXIT AFTER AN ERROR HALT
/SLOPE LOOP STARTING ADDRESS

/LGP GP TEST (MB>GP>AC) BINARY SEQUENCE M216,831

3403	4422	GPONT3, JMS I MAINT1	/SET NO PAUSE
3404	1004	TAD REGA	/GET DATA SOURCE
3405	7112	CLL RTR	/MOVE LSD INTO AC0,1,2
3406	7012	RTR	/MOVE
3407	0003	AND K7000	/SAVE AC00,01,02
3410	3005	DCA REGB	/STORE FOR TYPING
3411	1005	TAD REGB	/FETCH FOR TESTING
3412	3215	DCA ,*3	/SET FOR LGP GP
3413	6141	LINC	/LINC MODE
3414	0700	0700	/BASIC TAPE COMMAND
3415	0000	0000	/TEST LOCATION AC0,1,2
3416	0002	PDP	/GO TO 8 MODE
3417	1015	TAD K5400	/GET TAPE INST. TO AC
3420	6151	LMR	/LOAD MAINTENANCE REGISTER
3421	7300	CLA CLL	/CLEAR
3422	6154	XFR	/TRANSFER
3423	0070	AND K0007	/SAVE LGP AC0,1,2
3424	7112	RTR CLL	/SHIFT TO AC0,1,2
3425	7012	RTR	/SHIFT
3426	3006	DCA REGC	/STORE FOR TYPING
3427	1006	TAD REGC	/FETCH FOR TESTING
3430	7041	CIA	/NEGATE
3431	1005	TAD REGB	/SUBTRACT DATA SOURCE
3432	7050	SNA CLA	/TEST
3433	4503	JMS I NERROR	/NO ERROR
3434	4421	JMS I ERROR	/ERROR
3435	6163	GPMB	/MESSAGE 10
3436	7402	HLT	/ERROR HALT
3437	7610	SKP CLA	/EXIT
3440	3403	GPONT3	/SCOPE LOOP

/LGP GP TEST (MB>GP>AC) RANDOM SEQUENCE M216,931

3441	4422	GPONTR, JMS I MAINT1	/SET NOW PAUSE
3442	4432	JMS I RAN	/GENERATE A RANDOM NUMBER
3443	0003	AND K7000	/SAVE AC00,01,02
3444	3005	DCA REGB	/STORE FOR TYPING
3445	1005	TAD REGB	/FETCH IT
3446	3251	DCA ,*3	/STORE
3447	6141	LINC	/LINC MODE
3450	0700	0700	/BASIC TAPE
3451	0000	0000	/TEST WORD
3452	0002	PDP	/8 MODE
3453	1015	TAD K5400	/GET TAPE INST TO AC

3454	6151	LMR	/LOAD MAINTINANCE REGISTER
3455	7300	CLA CLL	/CLEAN
3456	6154	XFR	/TRANSFER
3457	0070	AND K0007	/SAVE SIGNIFICANT DATA
3460	7112	RTR CLL	/SHIFT INTO AC00,01,02
3461	7012	RTR	/SHIFT
3462	3000	DCA REGC	/STORE RESULTANT FOR TYPING
3463	1006	TAD REGC	/FETCH IT
3464	7041	CIA	/NEGATIVE
3465	1005	TAD REGB	/SUBTRACT SOURCE DATA
3466	7650	SNA CLA	/TEST AND CLEAN
3467	4503	JMS I NERROR	/TEST OKAY
3470	4421	JMS I ERROR	/TEST FAILED
3471	6163	GPMB	/MESSAGE TAG
3472	7402	HLT	/ERRON HALT
3473	7610	SKP CLA	/EXIT
3474	3441	GPONTR	/SCOPE LOOP

/DOES LGP GPCNT 0,1,2 COUNT NORMALLY M115 G25, PINS E1, F1, H1, J1

3475	4422	GPCNT5,	JMS I	MAINT1	/SET NO PAUSE
3476	7201		CLÄ IAC		/SET AC#0001
3477	3004		DCA	REGA	/SET MONITOR
3500	6141		LINC		/L MODE
3501	0701		0701		/TAPE
3502	0000		0000		/WASTE
3503	0002		PDP		/P MODE
3504	4547		JMS I	GPSET	/SETUP COUNT GPCNT GATE
3505	1004		TAD	REGA	/FETCH A TEST NUMBER
3506	0070		AND	K0007	/SAVE AC9,10,11
3507	3005		DCA	REGB	/STORE FOR TYPING
3510	1124		TAD	K6040	/SET UP FOR TP0
3511	6151		LMR		/GPCNT+1 > GPCNT
3512	7300		CLÄ CLL		/CLEAR AC,L
3513	6154		XFR		/READ GP COUNT
3514	0070		AND	K0007	/SAVE SIGNIFICANT DATA
3515	3006		DCA	REGC	/STORE FOR TYPING
3516	1006		TAD	REGC	/FETCH FOR TESTING
3517	7041		GIA		/NEGATE
3520	1005		TAD	REGB	/SUBTRACT DATA SOURCE
3521	7650		SNA CLA		/TEST
3522	4503		JMS I	NERROR	/NO ERROR
3523	4421		JMS I	ERROR	/ERROR
3524	6607		GPERS		/MESSAGE TAG
3525	7402		HLT		/ERROR HALT
3526	7610		SKP CLA		/EXIT
3527	3504		GPCNT5+7		/SCOPE LOOP

/DOES LIP MTP SETUP 0 LGP GP CNT M113 A36

3530	4422	GPCNT6,	JMS I	MAINT1	/SET NO PAUSE
3531	6141		LINC		/LINC MODE
3532	0701		0701		/TAPE
3533	7000		7000		/WASTED MEMORY SPACE
3534	0002		PDP		/P MODE
3535	4547		JMS I	GPSET	/SET UP TO COUNT
3536	1124		TAD	K6040	/SET UP FOR TP0
3537	6151		LMR		/+1*
3540	4422		JMS I	MAINT1	/SET NO PAUSE
3541	6141		LINC		/L MODE
3542	0700		0700		/GENERATE MTP SETUP
3543	7000		7000		/WASTE
3544	0002		PDP		/P MODE
3545	6154		XFR		/READ LGP GP CNT
3546	0070		AND	K0007	/SAVE SIGNIFICANT DATA
3547	3005		DCA	REGB	/STORE FOR TYPING
3550	1005		TAD	REGB	/FETCH
3551	7650		SNA CLA		/TEST RESULTS
3552	4503		JMS I	NERROR	/TEST OKAY
3553	4421		JMS I	ERROR	/TEST FAILED
3554	6176		GPERS		/MESSAGE TAG
3555	7402		HLT		/ERROR HALT

3550 7610
3557 3535

SKP CLM
GPCNT6+5

/EXIT
/SCOPE LOOP

/TEST COMPARE GATES GP CNT = 000 GP = 000

3560	4024	LGP000,	JMS	TPEPRE	/0 > LGP GP EQ GPC
3561	4422		JMS I	MAINT1	/SET NO PAUSE
3562	6141		LINC		/L MODE
3563	0705		0705		/SET LIN TINR 11(1)
3564	0777		0777		/WASTED MEMORY
3565	0002		POP		/P MODE
3566	1124		TAD	K0040	/MISC STATUS TO AC AND TP2
3567	6151		LMR		/LOAD MAINT REG
3570	7300		CLA CLL		/CLEAR AC,L
3571	6154		XFR		/READ DATA
3572	0043		AND	K0010	/SAVE GP EQ GPC BIT
3573	3005		DCA	REGB	/STORE IT FOR TYPING
3574	1005		TAD	REGB	/FETCH IT
3575	7640		SZA CLA		/TEST
3576	4503		JMS I	NERROR	/TEST OKAY
3577	4421		JMS I	ERROR	/TEST FAILED
3600	7309		LGM000		/MESSAGE TAG
3601	7402		HLT		/ERROR HALT
3602	7610		SKP CLA		/EXIT
3603	3560		LGP000		/SCOPE LOOP

/TEST COMPARE GATES GP CNT = 000 FLOAT A ONE THRU LGP GP

3604	4024	LGP001,	JMS	TPEPRE	/0 > LGP GP EQ GPC
3605	4422		JMS I	MAINT1	/SET NO PAUSE
3606	1063		TAD	K0400	/SET UP LGP GP NUMBER
3607	3005		DCA	REGB	/STORE IT
3610	1005		TAD	REGB	/FETCH THE FLOATING 1
3611	7104		CLL RAL		/SHIFT ONE LEFT
3612	7430		SZL		/NUMBER IS NOT INTO LINK YET
3613	5006		JMP	LGP001+2	/FLOATING 1 BUNK
3614	3005		DCA	REGB	/STORE FOR TYPING
3615	1005		TAD	REGB	/FETCH FOR TESTING
3616	3221		DCA	+3	/STORE IN TEST LOCATION
3617	6141		LINC		/L MODE
3620	0705		0705		/TINR 11(0), NOT
3621	0000		0000		/TEST ADDRESS
3622	0002		POP		/P MODE
3623	1124		TAD	K0040	/MISC STATUS TO AC AND TP2
3624	6151		LMR		/LOAD MAINT REG
3625	7300		CLA CLL		/CLEAR AC,L
3626	6154		XFR		/READ DATA
3627	0043		AND	K0010	/SAVE LGP GP EQ GPC
3630	3006		DCA	REGC	/STORE FOR TESTING
3631	1006		TAD	REGC	/FETCH FOR TESTING
3632	7690		SNA CLA		/WAS IT SET
3633	4503		JMS I	NERROR	/TEST OKAY
3634	4421		JMS I	ERROR	/TEST FAILED
3635	7373		LGM001		/MESSAGE TAG
3636	7402		HLT		/ERROR HALT
3637	7610		SKP CLA		/EXIT
3640	3610		LGP001+4		/SCOPE LOOP

/TEST COMPARE GATES GP CNT = 000 FLOAT A ONE THRU GP CNT

3641	4422	LGP002, JMS I	MAINT1	/SET NO PAUSE
3642	6141		LINC	/ L MODE
3643	0701		0701	/TAPE COMMAND & COUNTER
3644	0000		0000	/WASTED MEMORY
3645	0002		POP	/P MODE
3646	1062		TAD K1000	/GET TEST NUMBER
3647	3005		DCA REGB	/STORE
3650	4547	JMS I	GPSET	/SET UP COUNT GATE
3651	1124	TAD	K6040	/COUNT SET GP CNT = 001
3652	6151	LMR		/LOAD MAINT REG AND COUNT
3653	7300	CLA CLL		/CLEAR AC,L
3654	6154	XFR		/READ STATUS
3655	0043	AND	K0010	/SAVE EQUAL BIT
3656	1006	DCA	REGC	/STORE BIT
3657	1006	TAD	REGC	/FETCH FOR TESTING
3660	7640	SZA CLA		/TEST IT
3661	5322	JMP	LGB002	/BLUNDER
3662	1005	TAD	REGB	/GET B
3663	7104	RAL CLL		/SHIFT LEFT
3664	3005	DCA	REGB	/RESTORE
3665	4547	JMS I	GPSET	/SET GP CNT = 010
3666	1124	TAD	K6040	/DO THE COUNT AND LOAD MAINT IR
3667	6151	LMR		/LOAD MAINT REG
3670	7300	CLA CLL		/CLEAR AC,L
3671	6154	XFR		/READ STATUS
3672	0043	AND	K0010	/SAVE EQUAL BIT
3673	1006	TAD	REGC	/ADD TO REGB
3674	3005	DCA	REGC	/STORE IT
3675	1006	TAD	REGC	/FETCH FOR TESTING
3676	7640	SZA CLA		/TEST IT
3677	5322	JMP	LGB002	/BLUNDER
3700	1005	TAD	REGB	/GET B
3701	7104	RAL CLL		/SHIFT LEFT
3702	3005	DCA	REGB	/RESTORE
3703	4547	JMS I	GPSET	/SET UP FOR COUNT
3704	1124	TAD	K6040	/GENERATE TP0
3705	6151	LMR		/GP CNT = 011
3706	7300	CLA CLL		/CLEAR AC,L
3707	4547	JMS I	GPSET	/SET UP FOR COUNT
3710	1124	TAD	K6040	/GENERATE TP0
3711	6151	LMR		/GP CNT = 100
3712	7300	CLA CLL		/CLEAR AC,L
3713	6154	XFR		/READ STATUS
3714	0043	AND	K0010	/SAVE EQUAL BIT
3715	1006	TAD	REGC	/ADD C REGC
3716	3006	DCA	REGC	/STORE IN REGC
3717	1006	TAD	REGC	/FETCH IT
3720	7650	SNA CLA		/TEST
3721	4503	JMS I	NERROR	/TEST OKAY
3722	4421	LGB002, JMS I	ERROR	/TEST FAILED

3723	7373	LGM001	/MESSAGE TAG
3724	7402	HLT	/ERROR HALT
3725	7610	SKP CLA	/EXIT
3726	3641	LGP002	/SCOPE LOOP

/TC12-B-LTD TAPE DELAYS

/LTD TAPE FAIL DELAY TEST DOES/DID IT FIRE OFF M307 000

3727	1122	TFDLY1, TAD K4440	/GENERATE TP0 SHOULD ALREADY BE FIRED OFF TP0 IS JUST A SAFETY
3730	6151	LMR	/LOAD MAINTENANCE REGISTER
3731	7300	CLA CLL	/CLEAR
3732	6154	XPR	/READ STATUS
3733	7010	RAR	/MOVE AC11 TO LINK
3734	7030	SZL CLA	/TEST LINK
3735	4503	JMS I NERROR	/TEST OKAY
3736	4421	JMS I ERROR	/TEST FAILED
3737	6624	TFM1	/MESSAGE TAG
3740	7402	HLT	/ERROR HALT
3741	7610	SKP CLA	/EXIT
3742	3727	TFDLY1	/SCOPE LOOP

/LTD TAPE FAIL TEST DO WE GET A TAPE FAIL SIGNAL M111 021 (MAINT 0, IN PROGRESS 1)

3743	4824	TFDLY2, JMS TPEPRE	/0 - MAINT 0-MOTION
3744	1043	TAD K0010	/GET NO PAUSE
3745	6141	LINC	/L MODE
3746	0001	AOX	/SET NO PAUSE
3747	0707	0707	/SELECT UNIT 1 SET IN PROGRESS (1)
3750	7000	7000	/WASTE
3751	0002	PDP	/P MODE
3752	4545	JMS I TIMTF	/GO TO TAPE FAIL TIME
3753	4503	JMS I NERROR	/TEST OKAY
3754	4421	JMS I ERROR	/TEST FAILED
3755	6645	TFM2	/MESSAGE TAG
3756	7402	HLT	/ERROR HALT
3757	7610	SKP CLA	/EXIT
3760	3743	TFDLY2	/SCOPE LOOP

/LTD TAPE FAIL TEST (MAINT (1)) IN PROGRESS = 1)

3761	1076	TAD K0050	/GET CONSTANT TO KEEP MAINT SET
3762	3077	DCA KX0010	/SET UP CONSTANT
3763	4024	TFDLY3, JMS TPEPRE	/0>MOTION
3764	4422	JMS I MAINT1	/SET MAINT AND NO PAUSE
3765	6141	LINC	/L MODE
3766	0707	0707	/SET IN PROGRESS
3767	7000	7000	/WASTE
3770	0002	PDP	/P MODE
3771	4545	JMS I TIMTF	/GO TO TAPE FAIL DELAY TIME
3772	7610	SKP CLA	/INVERT
3773	4503	JMS I NERROR	/TEST OKAY
3774	4421	JMS I ERROR	/TEST FAILED
3775	6666	TFM3	/MESSAGE TAG
3776	7402	HLT	/ERROR HALT
3777	7610	SKP CLA	/EXIT
4000	3763	TFDLY3	/SCOPE LOOP
4001	1043	TAD K0010	/GET CONSTANT
4002	3077	DCA KX0010	/SET UP OLD CONSTANT

/LTD TAPE FAIL TEST (IN PROGRESS (0)) MAINT (0)

4003	3052	TFDLY4, DCA	K0040	/0 MAINT BIT
4004	4024	JMS TPEPRE		/0>MAINTENANCE, 0>MOTION, 0>IN PROGRESS
4005	4545	JMS I TIMTF		/GO TO TIME OUT ROUTINE
4006	7610	SKP CLA		/INVERT SENSE OF SKIP
4007	4503	JMS I NERROR		/TEST OKAY
4010	4421	JMS I ERROR		/TEST FAILED
4011	6707	TFDM3		/MESSAGE TAG
4012	7402	HLT		/ERROR HALT
4013	7610	SKP CLA		/EXIT
4014	4003	TFDLY4		/SCOPE LOOP
4015	1170	TAD	C0040	/GET MAINT BIT
4016	3052	DCA	K0040	/RESET MAINT BIT

/LTD ACIP DELAY TEST (DIRECTION (1)) (M307 C30) K1

4017	4024	ACIP1, JMS TPEPRE		/0 TO MOTION
4020	4422	JMS I MAINT1		/SET NO PAUSE
4021	1051	TAD K0020		/SET AC07
4022	6152	TRC		/0 TO DIRECTION
4023	7106	CLL RTL		/SET AC09
4024	6152	TRC		/1 TO DIRECTION START TIME OUT
4025	4532	JMS I TIMER		/GO TO ACIP TIMER
4026	4503	JMS I NERROR		/RETURN HERE IF TIMED OUT
4027	4421	JMS I ERROR		/RETURN HERE IF NOT TIMED OUT
4030	6730	ACIPT1		/MESSAGE TAG
4031	7402	HLT		/ERROR HALT
4032	7610	SKP CLA		/EXIT
4033	4034	ACIP2		/NO SCOPE LOOP

/LTD ACIP DELAY TEST (DIRECTION (0)) (M304 C17) M1

4034	1051	ACIP2, TAD K0020		/SET AC07
4035	6152	TRC		/0 TO DIRECTION
4036	4532	JMS I TIMER		/GO TO ACIP TIMER
4037	4503	JMS I NERROR		/RETURN HERE IF TIMED OUT
4040	4421	JMS I ERROR		/RETURN HERE IF NOT TIMED OUT
4041	6730	ACIPT2		/MESSAGE TAG
4042	7402	HLT		/ERROR HALT
4043	7610	SKP CLA		/EXIT
4044	4045	ACIP3		/NO SCOPE LOOP

/LTD ACIP DELAY TEST (MOTION (1)) (M304 C17) M1

4045	4422	ACIP3, JMS I MAINT1		/SET NO PAUSE
4046	6141	LINC		/L MODE
4047	0703	0703		/TAPE COMMAND
4050	7000	7000		/WASTE
4051	0002	PDP		/P MODE
4052	4532	JMS I TIMER		/GO TO ACIP TIMER
4053	4503	JMS I NERROR		/RETURN HERE IF TIMED OUT
4054	4421	JMS I ERROR		/RETURN HERE IF NOT TIMED OUT
4055	6771	ACIPT3		/MESSAGE TAG

4056 7402
4057 7610
4060 4017

HLT
SKP CLA
ACIP1

/ERROR HALT
/EXIT
/SCOPE LOOP ENTIRE TEST

/TAPE UNIT AND MOTION TC12-0-LMU
/
/LMU DIRECTION FLOP TEST (SET FORWARD)
/

4061	1051	DIRT1, TAD K0020	/SET BIT 7
4062	6152	TRC	/SET DIRT(0)
4063	7300	CLA CLL	/CLEAR AC,L
4064	1053	TAD K0100	/SET BIT 09
4065	6152	TRC	/SET DIRECTION FORWARD= (1)
4066	7300	CLA CLL	/CLEAR AC,L
4067	1037	TAD K5000	/UNITS MTN TO AC
4070	6151	LMR	/LOAD MAINTENANCE REGISTER
4071	7300	CLA CLL	/CLEAR AC,L
4072	6154	XFR	/READ UNITS > MTN
4073	0046	AND K0004	/SAVE DIRECTION
4074	7640	SEA CLA	/HAS DIRECTION SET
4075	4503	JMS I NERROR	/TEST OKAY
4076	4421	JMS I ERROR	/TEST FAILED
4077	7011	DIRM1	/MESSAGE TAG
4100	7402	HLT	/ERROR HALT
4101	7610	SKP CLA	/EXIT
4102	4061	DIRT1	/SCOPE LOOP

/LMU DIRECTION FLOP TEST (SET REVERSE) M121 C26 PIN F2,H2,J2
/

4103	1053	DIRT2, TAD K0100	/SET BIT 5
4104	6152	TRC	/SET DIRT(1)
4105	7300	CLA CLL	/CLEAR AC,L
4106	1051	TAD K0020	/SET BIT 07
4107	6152	TRC	/SET DIRECTION REVERSE (0)
4110	7300	CLA CLL	/CLEAR
4111	1037	TAD K5000	/UNITS MTN TO AC
4112	6151	LMR	/LOAD MAINTENANCE REGISTER
4113	7300	CLA CLL	/CLEAR AC,L
4114	6154	XFR	/READ UNITS MTN
4115	0046	AND K0004	/SAVE DIRECTION
4116	7650	SNA CLA	/HAS DIRECTION = 0
4117	4503	JMS I NERROR	/TEST OKAY
4120	4421	JMS I ERROR	/TEST FAILED
4121	7033	DIRM2	/MESSAGE TAG
4122	7402	HLT	/ERROR HALT
4123	7610	SKP CLA	/EXIT
4124	4103	DIRT2	/SCOPE LOOP

/LMU REVERSE DIRECTION MOTION 0 DLY 3 M121 C26 PIN D2,E2,J2
/

4125	1053	DIRT3, TAD K0100	/SET BIT 5
4126	6152	TRC	/SET DIRT(1)
4127	7300	CLA CLL	/CLEAR AC,L
4130	1053	TAD K0100	/SET BIT 5
4131	6152	TRC	/SET FORWARD
4132	4024	JMS TPEPRE	/0 - MOTION
4133	4422	JMS I MAINT1	/SET MAINT, PAUSE
4134	6141	LINC	/L MODE
4135	0703	0703	/GIVE TAPE COMMAND MTN DLY 3

4136 7000
 4137 0002
 4140 1037
 4141 6151
 4142 7300
 4143 6154
 4144 0046
 4145 7650
 4146 4503
 4147 4421
 4150 7055
 4151 7402
 4152 7010
 4153 4125

7000
 POP
 TAD K5000
 LMR
 CLA CLL
 XFR
 AND K0004
 SNA CLA
 JMS I NERROR
 JMS I ERROR
 DIRMS
 HLT
 SKP CLA
 DIRTS

/WASTE
 /P MODE
 /GET UNITS MTN > AG
 /LOAD MAINTENANCE REGISTER
 /CLEAR AC,L
 /READ DIRECTION
 /SAVE DIRECTION
 /WAS IT ZERO
 /TEST OKAY
 /TEST FAILED
 /MESSAGE TAG
 /ERROR HALT
 /EXIT
 /SCOPE LOOP

/LMU DIRECTION FLOP TEST M117 C23 PINS A1,B1,C1,D1,E1

```

DIRT4.  JMS I   MAINT1      /SET MAINT MODE
        LINC          /L MODE
        0700          /TAPE COMMAND TO SET IN PROGRESS
        7000          /WASTE
        PDP           /P MODE
        TAD          K0100  /SET BIT 05
        TRC          /SET FORWARD
        CLA CLL       /CLEAR AC,L
        TAD          K0016  /GET BLOCK MARK
        CMA CLL       /INVERT DATA
        JMS I   LOAD      /SET 0M
        TAD          K1000  /GET AC>TAD
        LMR           /LOAD MAINT REG
        CLA CLL CML RAR   /CLEAR AC,L SET AC00
        XFR           /SET TAC=4000
        CLA CLL       /0>AC,L
        TAD          K5040  /TP0,TP1,TP2.
        LMR           /SEARCH MODE
        CLA CLL       /CLEAR AC,L
        XFR           /READ DIRECTION
        AND          K0004  /SAVE DIRECTION
        SNA CLA       /TEST DIRECTION
        JMS I   NERROR    /TEST OK
        JMS I   ERROR    /TEST FAILED
        DIRM4        /MESSAGE TAG
        HLT          /ERROR HALT
        SKP CLA       /EXIT
        DIRT4        /SCOPE LOOP
    
```

/LMU DIRECTION FLOP TEST M117 C23 PINS A1,B1,C1,D1,E1 FALSE CONDITIONS

```

DIRT5.  JMS I   TPEPRE     /0>THE WORLD
        JMS I   MAINT1     /SET NO PAUSE
        LINC          /L MODE
        0700          /GIVE TAPE COMMAND
        7000          /WASTE
        PDP           /SET IN PROGRESS (1)
        TAD          K5040  /SET UP FOR TP1 PULSE
        LMR           /SEARCH
        CLA CLL       /0>AC,L
        TAD          K0016  /GET BLOCK MARK
        JMS I   LOAD      /LOAD WINDOW
        TAD          K5040  /GET UNITS MTN TO AC
        LMR           /TP2
        CLA CLL       /CLEAR AC,L
        TAD          K1000  /GET AC>TAD
        LMR           /LOAD MAINT REG
        CLA CLL       /CLEAR AC,L
        XFR           /SET TAC=7777.NOT
        TAD          K5040  /GET UNITS MTN TO AC
    
```

4233 6151
4234 7300
4235 6154
4236 0046
4237 7650
4240 4503
4241 4421
4242 7136
4243 7402
4244 7610
4245 4210

LMR
CLA CLL
XFR
AND K0004
SNA CLA
JMS I NERROR
JMS I ERROR
DIRMS
HLT
SNP CLA
DIRTS

/GENERATE TPO,1,2
/CLEAR AC,L
/READ DIRECTION
/SAVE DIRECTION
/TEST DIRECTION FLOP
/TEST OKAY
/TEST BAD
/MESSAGE TAG
/ERROR HALT
/EXIT
/SCOPE LOOP


```

4246 4422 DIRT6 JMS I MAINT1 /SET NO PAUSE
4247 6141 LINC /L MODE
4250 0703 0703 /MTB
4251 7000 7000 /WASTED MEMORY
4252 0002 PDP /I>PROGRESS
4253 1121 TAD K0016 /GET BLOCK MARK1
4254 4401 JMS I LOAD /SET MARK WINDOW TO BIN
4255 1062 TAD K1000 /GET AC>TAC
4256 6151 LMR /LOAD MAINTENANCE REGISTER
4257 7300 CLA CLL /CLEAR AC,L
4260 6154 XFR /TAC=0000
4261 1146 TAD K5040 /GET UNITS MTN + AC
4262 6151 LMR /LOAD MAINT IR
4263 7300 CLA CLL /CLEAR AC,L
4264 6154 XFR /READ DATA
4265 0046 AND K0004 /SAVE DIRECTION FLOP
4266 7640 SEA CLA /TEST
4267 4503 JMS I NERROR /TEST OKAY
4270 4421 JMS I ERROR /TEST FAILED
4271 7166 DIRM6 /MESSAGE TAG
4272 7402 HLT /ERROR HALT
4273 7610 SKP CLA /EXIT
4274 4246 DIRT6 /SCOPE LOOP

```

/LMU DIRECTION FLOP TEST N115, C27, PINS R1,S1,V1 (LTRF TAG 10 C1)

```

4275 4422 DIRT7 JMS I MAINT1 /SET MAINT MODE
4276 6141 LINC /L MODE
4277 0700 0700 /TAPE COMMAND
4300 7000 7000 /WASTED MEMORY
4301 0002 PDP /P MODE
4302 1121 TAD K0016 /GET BLOCK MARK
4303 4401 JMS I LOAD /SET MARK WINDOW TO BH
4304 1062 TAD K1000 /GET AC>TAC
4305 6151 LMR /LOAD MAINTENANCE REGISTER
4306 7300 CLA CLL /CLEAR AC,L
4307 1117 TAD K0002 /SET TAC 10(1)
4310 6154 XFR /SET TAC
4311 7300 CLA CLL /SET AC, L
4312 1146 TAD K5040 /GET UNITS + MTN TO AC
4313 6151 LMR /LOAD MAINT REQ GENERATE TP0
4314 7300 CLA CLL /CLEAR AC,L
4315 6154 XFR /READ DATA FROM TAPE CONTROL
4316 0046 AND K0004 /SAVE DIRECTION FLOP
4317 7640 SEA CLA /TEST DID IT CLEAR IN ERROR
4320 4503 JMS I NERROR /TEST OKAY
4321 4421 JMS I ERROR /TEST FAILED
4322 7211 DIRM7 /MESSAGE TAG
4323 7402 HLT /ERROR HALT
4324 7610 SKP CLA /EXIT
4325 4275 DIRT7 /SCOPE LOOP

```

/LMU DIRECTION FLOP TEST M117, B33, PINS, R2, B2, T2, U2, V2 AND M119, CR7 P1, S1, U1
 /FALSE CONDITIONS

4326	4422	DIRTS	JMS I	MAINT1	/SET MAINT AND NO PAUSE
4327	6141		LINC		/L MODE
4330	0703		0703		/WASTE SET IN PROGRESS
4331	7000		7000		/MTB
4332	0002		PDP		/P MODE
4333	1146		TAD	K9040	/TPB, TP1, TP2
4334	6191		LMR		/1>SEARCH
4335	7300		CLA	CLL	/CLEAR AC.L
4336	1121		TAD	K0016	/GET BLOCK MARK
4337	4401		JMS I	LOAD	/LOAD MARK WINDOW
4340	1062		TAD	K1000	/GET AC>TAG
4341	6191		LMR		/LOAD MAINTENANCE REGISTER
4342	7300		CLA	CLL	/CLEAR AC.L
4343	6194		XFR		/SET TAG=0000
4344	1146		TAD	K9040	/GET MOTION
4345	6191		LMR		/LOAD MAINT REGISTER
4346	7300		CLA	CLL	/CLEAR AC.L
4347	6194		XFR		/READ DATA
4350	7640		SEA	CLA	/TEST
4351	4903		JMS I	NERROR	/TEST OKAY
4352	4401		JMS I	ERROR	/TEST FAILED
4353	7236		DIRMS		/MESSAGE TAG
4354	7402		HLT		/ERROR HALT
4355	5176		JMP	GO	/START OVER AND DING THE BELL
4356	4326		DIRTS		/SCOPE LOOP

```

4400      4400      *4400
          /TAPE 5
          /SUBROUTINE TO LOAD A FIVE DIGIT BINARY NUMBER INTO
          /THE MARK WINDOW REGISTER NUMBER TO BE LOADED IS IN AC
          /JUSTIFIED RIGHT ON ENTRY INTO SUBROUTINE

4400 0000      LOADS, 0      /RETURN ADDRESS STORAGE
4401 7110      RAR CLL      /WORD TO BE LOADED IS MOVED RIGHT ONE BIT
4402 3104      DCA TEMPB   /STORE IT
4403 4422      JMS I MAINT1 /SET MAINTENANCE FLAG
4404 1061      TAD K7773    /SET AC=7773
4405 3007      DCA REGD     /SET TALLY TO MINUS 5
4406 1104      DOMORE, TAD TEMPB /GET STORED DATA
4407 0043      AND K0010    /SAVE CURRENT LOAD DATA BIT
4410 3024      DCA TPEPRZ   /STORE BIT TO BE LOADED
4411 1024      TAD TPEPRZ   /OR CLOCK PULSE AND DATA
4412 7040      CMA         /COMPLEMENT GET READY TO "OR"
4413 0051      AND K0020    /AND IN THE COMMAND BIT
4414 1024      TAD TPEPRE   /ADD IN LOADING BIT
4415 6151      LMR         /SEND IT
4416 7200      CLA         /CLEAR
4417 1104      TAD TEMPB   /MOVE NEXT MOST SIGNIFIED
4420 7004      RAL         /DATA BIT INTO LINK
4421 3104      DCA TEMPB   /STORE IT BACK
4422 2007      ISZ REGD     /DONE YET
4423 5206      JMP DOMORE   /NOT DONE YET
4424 7330      CLA CLL CML RAR /SET AC TO 4000
4425 6151      LMR         /LOAD MAINTENANCE REGISTER
4426 7200      CLA         /CLEAR ACCUMULATOR
4427 6154      XFR         /GET DATA
4430 3005      DCA REGB     /STORE FOR TYPING
4431 5600      JMP I LOADS  /EXIT

```

```

4600      *4600
/
/NON ERROR MONITOR DETERMINES IF OPERATOR WANTS TO LOOP ON NON FAILING TEST
NERROS, 0 /RETURN ADDRESS
4601 0000 JMS I MAINT1 /SET MAINTENANCE FLAG
4602 4422 CLA CLL IAC RTL /SET AC = 4
4603 7307 TAD NERROS /GET RETURN ADDRESS
4604 1200 DCA NERROS /UPDATE RETURN ADDRESS
4605 3200 TAD I NERROS /GET SCOPE LOOP ADDRESS
4606 1600 DCA ERRORS /STORE IT
4607 3221 ISZ REGA /UPDATE DATA
4610 2004 JMP I ERRORS /EXIT
4611 5621 LAS /READ SWITCHES
4612 7604 AND K0400 /SAVE SRS
4613 0063 SBA CLA /TEST AND CLEAR
4614 7640 JMP I ERRORS / LOOPING
4615 5621 CMA /SET AC=-1
4616 7040 TAD NERROS /ADD NERROS
4617 1200 DCA NERROS /STORE IN NERROS
4620 3200 JMP I NERROS /JUMP INDIRECT LOOP

/ERROR PROCESSOR, SCOPE LOOP, HALT, PRINT
ERRORS, 0 /RETURN ADDRESS STORAGE
4621 0000 LAS /READ SWITCHES
4622 7604 RAL /MOVE SR1 INTO AC00
4623 7004 SMA CLA /IS IT SET
4624 7700 JMP ASCII /NO TYPE A MESSAGE
4625 5254 JMS I BELL /RING THE BELL
4626 4530 ASCRXT, TAD ERRORS /GET CURRENT ERROR ADDRESS
4627 1221 CIA /INVERT IT
4630 7041 DCA LSTERR /STORE IN LAST ERROR
4631 3033 ISZ ERRORS /YES INDEX ESCAPE
4632 2221 LAS /READ SWITCHES
4633 7604 SMA CLA /IS SR0 SET
4634 7700 JMP I ERRORS /NO JUMP TO ERROR HALT
4635 5621 ISZ ERRORS /YES INDEX ESCAPE TO JUMP OUT
4636 2221 ISZ ERRORS /INDEX ERRORS TO SCOPE MODE
4637 2221 TAD I ERRORS /GET SXOPE ADDRESS
4640 1621 DCA NERROS /STORE IN TYPE
4641 3200 LAS /READ SWITCHES
4642 7604 RTL /MOVE SR02 TO AC0
4643 7006 SPA CLA /IS SCOPE MODE SELECTED
4644 7710 JMP I NERROS /YES CONTINUE IN SCOPE LOOP
4645 5600 ISZ REGA /UPDATE DATA
4646 2004 JMP I NERROS /TEST WITH NEW DATA
4647 5600 CMA /NO SET AC=7777
4650 7040 TAD ERRORS /SUBTRACT ONE FROM ERRORS
4651 1221 DCA ERRORS /STORE SELECTED ADDRESS
4652 3221 JMP I ERRORS /EXIT TO NEXT TEST
4653 5621

```

4654	7240	ASCII,	CLA CMA	/SET C(AC)=1
4655	1621		TAD I ERRORS	/GET MESSAGE ADDRESS STORAGE
4656	3012		DCA PINT	/STORE IT IN AUTO INDEX REGISTER
4657	1221		TAD ERRORS	/GET RETURN ADDRESS
4660	1033		TAD LSTERR	/SUBTRACT LAST ERROR ADDRESS
4661	7650		SNA CLA	/TEST
4662	5573		JMP I DATYPA	/SAME GO TYPE DATA
4663	1412		TAD I PINT	/GET FIRST CHARACTER
4664	3200		DCA NERROS	/SAVE IT
4665	1200		TAD NERROS	/GET IT
4666	7450		SNA	/TEST IT
4667	5227		JMP ASCRXT	/NUMBER=EXIT
4670	7040		CMA	/INVERT IT
4671	7450		SNA	/NUMBER=EXITA
4672	5320		JMP DATUM	/TYPE OUT DATA ROUTINE
4673	7040		CMA	/CHANGE IT BACK
4674	7112		RTR CLL	/SWAP AC TO THE RIGHT
4675	7012		RTR	/MOVE
4676	7012		RTR	/MOVE
4677	4303		JMS TYPECH	/TYPE IT
4700	1200		TAD NERROS	/GET IT AGAIN
4701	4303		JMS TYPECH	/TYPE IT
4702	5263		JMP ASCII+7	/MUST BE MORE WORDS THAT NEED TYPING
4703	0000	TYPECH,	0	
4704	0042		AND K0077	/SAVE SIGNIFICENT PART
4705	3034		DCA SPACE	/STORE WORD
4706	1034		TAD SPACE	/FETCH IT
4707	7650		SNA CLA	/TEST FOR 00 CRLF CODE
4710	4571		JMS I CRLFA	/YES IT WAS
4711	1034		TAD SPACE	/NO TYPE IT
4712	1106		TAD M40	/SUBTRACT 40
4713	7510		SPA	/TEST POLARITY
4714	1053		TAD K0100	/ADD 340
4715	1040		TAD K240	/ADD 240
4716	4423		JMS I TYPE	/TYPE
4717	5703		JMP I TYPECH	/EXIT
4720	1412	DATUM,	TAD I PINT	/GET ADDRESS OF REGISTER
4721	3200		DCA NERROS	/STORE IN TEMP
4722	1200		TAD NERROS	/GET TEMP
4723	7650		SNA CLA	/TEST FOR EXIT
4724	5227		JMP ASCRXT	/EQUALS 0000 EXIT
4725	1600		TAD I NERROS	/GET DATA
4726	4332		JMS OCTYP	/TYPE IT
4727	1040		TAD K240	/SPACE
4730	4423		JMS I TYPE	/TYPE IT
4731	5320		JMP DATUM	/TYPE NUMERIC DATA
4732	0000	OCTYP,	0	/RETURN ADDRESS STORAGE
4733	3303		DCA TYPECH	/STORE DATA TO BE PRINTED
4734	1123		TAD K7774	/SET UP TALLY
4735	3034		DCA SPACE	/SET IT

4736	1041	HERE,	TAD K1026	/GET FLAG NUMBER
4737	3571	REDO,	DCA I CRLFA	/STORE
4740	1303		TAD TYPECH	/FETCH DATA
4741	7004		RAL	/SHIFT INTO LINK
4742	3303		DCA TYPECH	/STORE IT
4743	1571		TAD I CRLFA	/FETCH FLAG
4744	7004		RAL	/SHIFT INTO LINK
4745	7420		SNL	/TEST LINK FOR DONE FLAG
4746	5572		JMP I REDOA	/NOT DONE
4747	4423		JMS I TYPE	/DONE TYPE IT
4750	2034		ISZ SPACE	/DONE WITH ENTIRE WORD YET
4751	5336		JMP HERE	/NOPE MORE CHARACTERS
4752	5732		JMP I OCTYP	
4753	0000	CRLF,	0	/RETURN ADDRESS STORAGE
4754	1055		TAD K0215	/GET CR
4755	4423		JMS I TYPE	/TYPE IT
4756	1050		TAD K0212	/GET LF
4757	4423		JMS I TYPE	/TYPE IT
4760	1056		TAD K0177	/SET TO RUBOUT
4761	5753		JMP I CRLF	/EXIT
4762	1412	DATYP,	TAD I PINT	/GET A TERM OFF OF TYPE LIST
4763	7450		SNA	/END OF LIST?
4764	5501		JMP I ASCRA	
4765	7040		CHA	/INVERT
4766	7640		SEA CLA	/BEGINNING OF DATA
4767	5362		JMP DATYP	/NO
4770	4571		JMS I CRLFA	/YES OK RETURN THE TTY CARRIAGE AND LINE FEED
4771	7300		CLA CLL	/CLEAR AC AND LINK
4772	5500		JMP I DATUMA	

```

      5000      *5000
3000  0000      /ROUTINE TO SET MAINTENANCE FLOP
3001  7200      MAINTS, 0
3002  1052      CLA /RETURN ADDRESS STORAGE
3003  1043      TAD K0040 /CLEAR AC
3004  6141      TAD K0010 /SET BIT 5
3005  0001      LINC /SET MAINTANENCE MODE
3006  0002      AOX /GO TO LINC MODE
3007  7200      PDP /LOAD EXTENDED OPERATIONS REGISTER
3010  5600      CLA /GO TO PDP-8 MODE
           JMP I MAINTS /CLEAR AC
           /EXIT

3011  0000      BELLS, 0 /RING THE BELL
3012  7604      LAS
3013  0053      AND K0100
3014  7640      SEA CLA
3015  5611      JMP I BELLS
3016  1105      TAD K0207
3017  4423      JMS I TYPE
3020  5611      JMP I BELLS
3021  7000      NOP

```

```

5022 0000  RANDM, 0
5023 1247  TAD RNA /RANDOM NUMBER GENERATOR
5024 1250  TAD RNB
5025 1251  TAD RNC
5026 3247  DCA RNA
5027 7004  RAL
5030 1247  TAD RNA
5031 1250  TAD RNB
5032 1251  TAD RNC
5033 3250  DCA RNB
5034 7004  RAL
5035 1247  TAD RNA
5036 1250  TAD RNB
5037 1251  TAD RNC
5040 3251  DCA RNC
5041 7004  RAL
5042 1247  TAD RNA
5043 3247  DCA RNA
5044 1250  TAD RNB
5045 1251  TAD RNC
5046 5622  JMP I RANDM
5047 0001  RNA, 0001 /RANDOM NUMBER SEED 1
5050 0000  RNB, 0000 /SEED 2
5051 0000  RNC, 0000 /SEED 3

5052 0000  TYP0UT, 0 /RETURN ADDRESS STORAGE
5053 6046  TIS /PRINT
5054 6041  TSP /WAIT FOR FLAG
5055 5254  JMP ,-1 /WAIT
5056 7300  CLA CLL /CLEAR AC, L
5057 5632  JMP I TYP0UT /EXIT

```


5060	6141	LOGC,	LINC	
5061	0700		0700	
5062	0000		0000	
5063	0002		PDP	
5064	5540		JMP I	PNTE

5065	0000	LMRWBS, 0		/SUBROUTINE TO LOAD RWD VIA MAINT GATE
5066	7140	CLL CMA		
5067	7012	RTR		
5070	7010	RAR		
5071	3011	DCA REGF	/STORE IT	
5072	1123	TAD K7774	/SET 7774	
5073	3104	DCA TEMPB		
5074	4422	JMS I MAINT1	/SET MAINT AND NO PAUSE	
5075	7010	RAR	/MOVE LINK AC09	
5076	3024	DCA TPEPRE	/STORE IT	
5077	1011	TAD REGF		
5100	0116	AND K0001		
5101	3010	DCA REGE	/STORE LSD QUAD 8,9,10,11	
5102	1011	TAD REGF		
5103	7012	RTR		
5104	7010	RAR		
5105	3007	DCA REGD		
5106	1007	TAD REGD		
5107	0117	AND K0002		
5110	1010	TAD REGE	/STORE LSD 2	
5111	3010	DCA REGE		
5112	1007	TAD REGD		
5113	7012	RTR		
5114	7010	RAR		
5115	0046	AND K0004		
5116	1010	TAD REGE		
5117	1035	TAD K2000		
5120	6102	TRC		
5121	7300	CLL CLL		
5122	1024	TAD TPEPRE		
5123	7004	RAL	/RESTOR LINK	
5124	1011	TAD REGF		
5125	7004	RAL		
5126	3011	DCA REGF		
5127	2104	ISZ TEMPB		
5130	5274	JMP LMRWBS+7		
5131	5665	JMP I LMRWBS		

/ACIP DELAY TIMER SUBROUTINE

```

/
TIME, 0000 /RETURN ADDRESS STORAGE
5132 0000
5133 7240 CLA CMA /SET AC=7777
5134 3004 DCA REGA /SET MONITOR
5135 1036 TAD K4400 /GET STATES N TIMING TO AC
5136 6151 LMR /LOAD MAINTENANCE REGISTER
5137 7300 CLA CLL /CLEAR AC,L
5140 3011 DCA REGF
5141 1044 TAD K7760
5142 3010 DCA REGE
5143 6154 XFR
5144 0043 AND K0010
5145 3005 DCA REGB
5146 1005 TAD REGB
5147 7650 SNA CLA
5150 5363 JMP BADXIT
5151 6154 XFR
5152 0043 AND K0010
5153 3005 DCA REGB
5154 1005 TAD REGB
5155 7640 SZA CLA
5156 5732 JMP I TIME
5157 2011 ISZ REGF
5160 5351 JMP , -7
5161 2010 ISZ REGE
5162 5360 JMP , -2
5163 2332 BADXIT, ISZ TIME /WAIT
5164 5732 JMP I TIME /SET RETURN ADDRESS TO ERROR MONITOR
/EXIT TO ERROR MONITOR
*5200
/

```

/TAPE FAIL SUBROUTINE

```

/
TFSUB, 0000 /RETURN ADDRESS STORAGE
5200 0000
5201 7300 CLA CLL /CLEAR AC,L
5202 1051 TAD K0020 /GET REV
5203 6192 TRC /SET REV
5204 7300 CLA CLL /CLEAR AC,L
5205 1121 TAD K0016 /GET BM
5206 4401 JMS I LOAD /SET BM
5207 1051 TAD K0020 /GEN TP3,TP4
5210 6151 LMR /GEN TP4
5211 7300 CLA CLL /CLEAR AC,L
5212 4401 JMS I LOAD /LOAD WINDOW
5213 1170 TAD G0040 /GET TP1
5214 6151 LMR /0 MOTION
5215 7300 CLA CLL /CLEAR AC,L
5216 1077 TAD KX0010
5217 6141 LINC /L MODE
5220 0001 AOX /SET NO PAUSE
5221 0002 POP /P MODE
5222 7240 CLA CMA /SET AC=7777
5223 3004 DCA REGA /SET FOR 1 CYCLE EXIT

```

5224 3010
 5225 1044
 5226 3011
 5227 1037
 5230 6151
 5231 7300
 5232 6154
 5233 4043
 5234 7640
 5235 5600
 5236 2010
 5237 5232
 5240 2011
 5241 5237
 5242 2200
 5243 5600

DCA REGE
 TAD K7760
 DCA REGF
 TAD K9000
 LMR
 CLA CLL
 XFR
 AND K0010
 SZA CLA
 JMP I TFSUB
 ISE REGE
 JMP ,=5
 ISE REGF
 JMP ,=R
 ISE TFSUB
 JMP I TFSUB

/0>REGE
 /0>REGF
 /GET STATES TIMING AC
 /LOAD MAINT REG
 /0>AC,L
 /READ DATA
 /SAVE MOTION
 /IS IT CLEARED YET
 /YES, EXIT TO
 /WAIT
 /UPDATE TO ERROR EXIT
 /EXIT TO INPUT+2

/SET GROUP COUNTER UP

```

GPSETS, 0000
5244 0000 /RETURN ADDRESS STORAGE
5245 1062 /GET AC>TAC
5246 6151 /LOAD MAINT REG
5247 7240 /SET AC=7777
5250 6154 /SET TAC=7777
5251 4422 /SET MAINT MODE
5252 1051 /GET REVERSE
5253 6152 /SET DIRECTION (0) REV
5254 7300 /CLEAR AC, L
5255 1121 /SET AC =BM
5256 4401 /SET MARK WINDOW TO CHECK WINDOW
5257 1053 /GET FWD DIRECTION BIT
5260 6152 /SET FWD
5261 7300 /CLEAR AC, L
5262 1052 /GENERATE TP0, TP1, TP2 1 > SEARCH
5263 6151 /BLOCK MODE
5264 7300 /CLEAR AC, L
5265 1051 /SET DIR REV
5266 6152 /SET DIRECTION=0
5267 7300 /CLEAR AC, L
5270 4401 /07 WINDOW
5271 1052 /GENERATE TP0
5272 6151 /TP0, TP1, TP2
5273 7301 /SET CM
5274 4401 /SET MARK WINDOW TO CM
5275 1124 /GENERATE TP0, TP1, TP2
5276 6151 /GENERATE TP0, TP1, TP2 SET CHK WRD
5277 7300 /CLEAR AC, L
5300 6154 /READ GROUP COUNTER (WHATEVER IT IS)
5301 0070 /SAVE GROUP COUNTER BITS
5302 3011 /STORE
5303 1004 /GET REGA
5304 7640 /FIRST PASS?
5305 5310 /EXIT
5306 1011 /YES GET GP CNT
5307 7300 /SYNC IN MONITOR
5310 1062 /GET AC>TAC
5311 6151 /LOAD MAINT REG
5312 7240 /SET AC=7777
5313 6154 /SET TAC
5314 7300 /CLEAR AC, L
5315 5644 /EXIT

```

5316	0014	CHECK,	0014	/LWN CM
5317	2716		2716	/0000
5320	4003		4003	
5321	1500		1500	
5322	7777		EXITA	
5323	0000		REGB	
5324	0000		EXIT	
5325	0014	GUARD,	0014	/LWN CM
5326	2716		2716	/0000
5327	4007		4007	
5330	1500		1500	
5331	7777		EXITA	
5332	0000		REGB	
5333	0000		EXIT	
5334	0014	DATA,	0014	/LWN DM
5335	2716		2716	/0000
5336	4004		4004	
5337	1500		1500	
5340	7777		EXITA	
5341	0000		REGB	
5342	0000		EXIT	
5343	0014	FINAL,	0014	/LWN FM
5344	2716		2716	/0000
5345	4006		4006	
5346	1500		1500	
5347	7777		EXITA	
5350	0000		REGB	
5351	0000		EXIT	
5352	0014	BLOCK,	0014	/LWN BM
5353	2716		2716	/0000
5354	4001		4001	
5355	1500		1500	
5356	7777		EXITA	
5357	0000		REGB	
5360	0000		EXIT	
5361	0014	INTERM,	0014	/LWN IM
5362	2716		2716	/0000
5363	4011		4011	
5364	1500		1500	
5365	7777		EXITA	
5366	0000		REGB	
5367	0000		EXIT	
5370	0014	END,	0014	/LWN EM
5371	2716		2716	/0000
5372	4005		4005	
5373	1500		1500	
5374	7777		EXITA	
5375	0000		REGB	
5376	0000		EXIT	

5377	0014	THAC,	0014	/LTR TMA AG FAILED
5400	2422		2422	/AG TMA
5401	4024		4024	/0000 0000
5402	1501		1501	
5403	4001		4001	
5404	0340		0340	
5405	4006		4006	
5406	0111		0111	
5407	1405		1405	
5410	0440		0440	
5411	0001		0001	
5412	0340		0340	
5413	4040		4040	
5414	2415		2415	
5415	0100		0100	
5416	7777		EXITA	
5417	0005		REGB	/DATA SENT TO TMA
5420	0006		REGB	/DATA RECEIVED FROM TMA
5421	0000		EXIT	
5422	0014	MAINTC,	0014	/LTMR MAINT IR TAPE PRESET
5423	2415		2415	/0000 0000 0000
5424	2240		2240	
5425	1501		1501	
5426	1116		1116	
5427	2440		2440	
5430	1122		1122	
5431	4024		4024	
5432	0120		0120	
5433	0340		0340	
5434	2022		2022	
5435	0523		0523	
5436	0524		0524	
5437	4000		4000	
5440	7777		EXITA	
5441	0005		REGB	/DATA SENT TO LTMR MAINT IR
5442	0004		REGA	/DATA SENT TO TB
5443	0006		REGB	/DATA RXED FROM TAPE BUFFER
5444	0000		EXIT	
5445	0014	MAINTI,	0014	/LTMR IOT DECODER
5446	2415		2415	/0000 0000
5447	2240		2240	
5450	1117		1117	
5451	2440		2440	
5452	0405		0405	
5453	0317		0317	
5454	0405		0405	
5455	2200		2200	
5456	7777		EXITA	
5457	0004		REGA	/DATA SENT TO TB
5460	0005		REGB	/DATA RECEIVED FROM TB
5461	0000		EXIT	

5462 0014 MAINT4, 0014
 5463 2415 2415
 5464 2240 2240
 5465 1117 1117
 5466 2440 2440
 5467 6661 6661
 5470 6564 6564
 5471 4000 4000
 5472 7777 EXITA
 5473 0133 K0252
 5474 0005 REGB
 5475 0000 EXIT

/LTHR IOT 0154
 /5252 0000

/DATA PUT INTO TB
 /DATA RECEIVED BACK

5476 0014 MAINTM, 0014
 5477 2415 2415
 5500 2240 2240
 5501 1501 1501
 5502 1116 1116
 5503 2440 2440
 5504 1517 1517
 5505 0405 0405
 5506 4006 4006
 5507 1417 1417
 5510 2000 2000
 5511 7777 EXITA
 5512 0005 REGB
 5513 0006 REGC
 5514 0000 EXIT

/LTHR MAINT MODE FLOP
 /0000 0000

/DATA SENT TO MAINT MODE FLOP
 /DATA RECEIVED BACK

5515 0014 MAINTP, 0014
 5516 2415 2415
 5517 2240 2240
 5520 1501 1501
 5521 1116 1116
 5522 2440 2440
 5523 1517 1517
 5524 0405 0405
 5525 4024 4024
 5526 0120 0120
 5527 0540 0540
 5530 2022 2022
 5531 0523 0523
 5532 0524 0524
 5533 4000 4000
 5534 7777 EXITA
 5535 0005 REGB
 5536 0006 REGC
 5537 0000 EXIT

/LTHR MAINT MODE TAPE PRESET
 /0000 0000

/DATA SENT TO MAINT MODE FLOP
 /DATA RECEIVED FROM MAINT MODE FLOP

5540 0014 MAINT2, 0014
5541 2415 2415
5542 2240 2240
5543 1117 1117
5544 2440 2440
5545 6661 6661
5546 6562 6562
5547 4000 4000
5550 7777 EXITA
5551 0005 REG8
5552 0000 EXIT

/LTMR IOT 6152
/0000

/DATA IN MAINT MODE FLOP

5553 0014 LIN77, 0014
5554 1116 1116
5555 4024 4024
5556 0103 0103
5557 4005 4005
5560 2140 2140
5561 6767 6767
5562 6767 6767
5563 4050 4050
5564 2401 2401
5565 0375 0375
5566 7777 EXITA
5567 0005 REG8
5570 0000 EXIT

/LIN TAC EQ 7777 (TAC=7777)
/

/DATA IN TAC REGISTER

5571	0014	LINIPS, 0014	/LIN IIVR TAPE PRESET
5572	1116	1116	
5573	4024	4024	
5574	1116	1116	
5575	2240	2240	
5576	2401	2401	
5577	2005	2005	
5600	4020	4020	
5601	2205	2205	
5602	2305	2305	
5603	2400	2400	
5604	7777	EXITA	
5605	0005	REGB	/DATA SENT TO INSTRUCTION REGISTER
5606	0006	REGC	/DATA RECEIVED FROM INSTRUCTION DECODER
5607	0000	EXIT	
5610	0014	LININD, 0014	/LIN INSTRUCTION DECODER
5611	1116	1116	/0000 0000
5612	4011	4011	
5613	1623	1623	
5614	2422	2422	
5615	2503	2503	
5616	2411	2411	
5617	1716	1716	
5620	4004	4004	
5621	0503	0503	
5622	1704	1704	
5623	0522	0522	
5624	4000	4000	
5625	7777	EXITA	
5626	0005	REGB	/DATA SENT TO INSTRUCTION REGISTER
5627	0006	REGC	/DATA RECEIVED FROM INSTRUCTION DECODER
5630	0000	EXIT	
5631	0014	LINIDT, 0014	/LIN I BIT
5632	1116	1116	/0000 0000
5633	4011	4011	
5634	4002	4002	
5635	1124	1124	
5636	4000	4000	
5637	7777	EXITA	
5640	0005	REGB	/DATA SENT TO I BIT
5641	0006	REGC	/DATA RECEIVED FROM I BIT
5642	0000	EXIT	

5643	0014	TBWAC,	0014
5644	2422		2422
5645	4024		4024
5646	0216		0216
5647	4001		4001
5650	0340		0340
5651	0601		0601
5652	1114		1114
5653	0504		0504
5654	4000		4000
5655	0103		0103
5656	4040		4040
5657	4024		4024
5660	0216		0216
5661	4000		4000
5662	7777		EXITA
5663	0005		REGB
5664	0006		REGC
5665	0000		EXIT

/LTR TBN AC FAILED

/AC TBN

/0000 0000

/DATA SENT TO TBN

/DATA RECEIVED FROM TBN

5666	0014	TBAC,	0014
5667	2422		2422
5670	4024		4024
5671	0240		0240
5672	0103		0103
5673	4000		4000
5674	0111		0111
5675	1405		1405
5676	0400		0400
5677	0103		0103
5700	4040		4040
5701	4024		4024
5702	0200		0200
5703	7777		EXITA
5704	0005		REGB
5705	0006		REGC
5706	0000		EXIT
5707	0014	TACAC,	0014
5710	2422		2422
5711	4024		4024
5712	0103		0103
5713	4001		4001
5714	0340		0340
5715	0601		0601
5716	1114		1114
5717	0504		0504
5720	0001		0001
5721	0340		0340
5722	4040		4040
5723	2401		2401
5724	0300		0300
5725	7777		EXITA
5726	0005		REGB
5727	0006		REGC
5730	0000		EXIT

/LTR TB AC FAILED
/AC TB
/0000 0000

/DATA SENT TO TB
/DATA RECEIVED FROM TB

/LTR TAC AC FAILED
/AC TAC
/0000 0000

/DATA SENT TO TAC
/DATA RECEIVED FROM TAC

5731	0014	TBADD,	0014
5732	2422		2422
5733	4024		4024
5734	0240		0240
5735	2401		2401
5736	0340		0340
5737	0104		0104
5740	0440		0440
5741	0601		0601
5742	1114		1114
5743	0504		0504
5744	4000		4000
5745	2402		2402
5746	4040		4040
5747	4024		4024
5750	0103		0103
5751	4040		4040
5752	0717		0717

/LTR TB TAC ADD FAILED

/TB TAC GOOD ACTUAL

5753	1704	1704
5754	4001	4001
5755	0324	0324
5756	2501	2501
5757	1400	1400
5760	7777	EXITA
5761	0005	REGB
5762	0006	REGC
5763	0007	REGD
5764	0010	REGE
5765	0000	EXIT

5766	0014	DATABD,	0014
5767	2716		2716
5770	4015		4015
5771	0122		0122
5772	1340		1340
5773	2711		2711
5774	1604		1604
5775	1727		1727
5776	4004		4004
5777	1122		1122
6000	5060		5060
6001	5100		5100
6002	7777		EXITA
6003	0010		REGE
6004	0005		REG0
6005	0000		EXIT

/LWN MARK WINDOW DIR (0)
/0000 0000

6006	0014	DATAFW,	0014
6007	2716		2716
6010	4015		4015
6011	0122		0122
6012	1340		1340
6013	2711		2711
6014	1604		1604
6015	1727		1727
6016	4004		4004
6017	1122		1122
6020	5061		5061
6021	5100		5100
6022	7777		EXITA
6023	0010		REGE
6024	0005		REG0
6025	0000		EXIT

/LWN MARK WINDOW DIR (1)
/0000 0000

6026	0014	TMASET, 0014
6027	2422	2422
6030	4024	4024
6031	1501	1501
6032	4023	4023
6033	0524	0524
6034	2520	2520
6035	4001	4001
6036	0340	0340
6037	0601	0601
6040	1114	1114
6041	0504	0504
6042	0001	0001
6043	0340	0340
6044	4040	4040
6045	2415	2415
6046	0140	0140
6047	2305	2305
6050	2425	2425
6051	2000	2000
6052	7777	EXITA
6053	0005	REGB
6054	0006	REGC
6055	0000	EXIT

/LTR TMA SETUP AC FAILED
/AG TMA SETUP
/0000 0000

/DATA SENT TO TMA SETUP
/DATA RECEIVED FROM TMA SETUP

6056	0014	RWBSHF,	0014	/LTR RWB SHIFT
6057	2422		2422	/0000 0000
6060	4022		4022	
6061	2702		2702	
6062	4023		4023	
6063	1011		1011	
6064	0624		0624	
6065	4000		4000	
6066	7777		EXITA	
6067	0005		REGB	/ACTUAL DATA
6070	0006		REGC	/SIMULATED CORRECT DATA
6071	0000		EXIT	
6072	0014	UNIDAT,	0014	/LTC UNIT SELECT
6073	2403		2403	
6074	4025		4025	
6075	1611		1611	/0000 0000 0000
6076	2440		2440	
6077	2305		2305	
6100	1405		1405	
6101	0324		0324	
6102	4000		4000	
6103	7777		EXITA	
6104	0005		REGB	
6105	0006		REGC	
6106	0007		REGD	
6107	0000		EXIT	
6110	0014	UNMES,	0014	/LTC UNIT SELECT PRESET
6111	2403		2403	
6112	4025		4025	
6113	1611		1611	/0000
6114	2440		2440	
6115	2305		2305	
6116	1405		1405	
6117	0324		0324	
6120	4020		4020	
6121	2205		2205	
6122	2305		2305	
6123	2400		2400	
6124	7777		EXITA	
6125	0005		REGB	
6126	0000		EXIT	

6127	0014	GPEDB,	0014	/LGD GR=GPO INR 11
6130	0720		0720	
6131	4007		4007	
6132	2275		2275	/0000 0000
6133	0720		0720	
6134	0340		0340	
6135	1116		1116	
6136	2240		2240	
6137	6161		6161	
6140	4006		4006	
6141	0111		0111	
6142	1405		1405	
6143	0400		0400	
6144	7777		EXITA	
6145	0005		REGB	
6146	0006		REGC	
6147	0000		EXIT	
6150	0014	GPPRE,	0014	/LGP GP=GPC PRESET
6151	0720		0720	
6152	4007		4007	
6153	2075		2075	
6154	0720		0720	
6155	0340		0340	
6156	2022		2022	
6157	0523		0523	
6160	0524		0524	
6161	4000		4000	
6162	0000		EXIT	
6163	0014	GPMB,	0014	/LGP GRP DATA
6164	0720		0720	/0000 0000
6165	4007		4007	
6166	2220		2220	
6167	4004		4004	
6170	0124		0124	
6171	0100		0100	
6172	7777		EXITA	
6173	0005		REGB	
6174	0006		REGC	
6175	0000		EXIT	
6176	0014	GPERS,	0014	/LGP MTP SETUP FAILED TO CLEAR COUNTER
6177	0720		0720	/0000
6200	4015		4015	
6201	2420		2420	
6202	4023		4023	
6203	0524		0524	
6204	2520		2520	
6205	4006		4006	
6206	0111		0111	
6207	1405		1405	
6210	0440		0440	
6211	2417		2417	
6212	4003		4003	
6213	1405		1405	
6214	0122		0122	

0215	4003	4003
0216	1725	1725
0217	1624	1624
0220	0522	0522
0221	4000	4000
0222	7777	EXITA
0223	0005	REOB
0224	0000	EXIT
0225	0014	0014
0226	2423	2423
0227	4014	4014
0230	0360	0360
0231	6140	6140
0232	0601	0601
0233	1114	1114
0234	0504	0504
0235	4024	4024
0236	1740	1740
0237	1116	1116
0240	1011	1011
0241	0211	0211
0242	2440	2440
0243	0405	0405
0244	0317	0317
0245	0411	0411
0246	1607	1607
0247	4017	4017
0250	0640	0640
0251	0515	0515
0252	4000	4000
0253	7777	EXITA
0254	0005	REOB
0255	0000	EXIT

LCOM,

/LTS LCB1 FAILED TO INHIBIT DECODING OF EM
/0000

6256	0014	LCXOTS,	0014	/LCX EXTEND OPS DATA XFER
6257	0330		0330	/0000 0000
6260	4005		4005	
6261	3024		3024	
6262	0516		0516	
6263	0440		0440	
6264	1720		1720	
6265	2340		2340	
6266	0401		0401	
6267	2401		2401	
6270	4030		4030	
6271	0605		0605	
6272	2200		2200	
6273	7777		EXITA	
6274	0005		REGB	/DATA SENT TO EXTEND OPS REG
6275	0006		REGC	/DATA RECEIVED FROM EXTEND OPS REG
6276	0000		EXIT	
6277	0014	LCXIPS,	0014	/LCX EXTEND OPS PRESET
6300	0330		0330	/0000
6301	4005		4005	
6302	3024		3024	
6303	0516		0516	
6304	0440		0440	
6305	1720		1720	
6306	2340		2340	
6307	2022		2022	
6310	0523		0523	
6311	0524		0524	
6312	4000		4000	
6313	7777		EXITA	
6314	0005		REGB	/DATA IN EXTENDED OPERATIONS REGISTER
6315	0000		EXIT	
6316	0014	LCXLTO,	0014	/LCX LOAD EXT OP REG
6317	0330		0330	/0000
6320	4014		4014	
6321	1701		1701	
6322	0440		0440	
6323	0530		0530	
6324	2440		2440	
6325	1720		1720	
6326	4022		4022	
6327	0507		0507	
6330	4000		4000	
6331	7777		EXITA	
6332	0005		REGB	/DATA IN EXTENDED OPERATIONS REGISTER
6333	0000		EXIT	

6334	0014	RWBMG,	0014
6335	2422		2422
6336	4001		4001
6337	0340		0340
6340	2227		2227
6341	0240		0240
6342	5026		5026
6343	1101		1101
6344	4015		4015
6345	0111		0111
6346	1624		1624
6347	4007		4007
6350	0124		0124
6351	0551		0551
6352	4006		4006
6353	0111		0111
6354	1405		1405
6355	0400		0400
6356	0103		0103
6357	4040		4040
6360	4022		4022
6361	2702		2702
6362	4000		4000
6363	7777		EXITA
6364	8805		REGB
6365	0006		REGC
6366	0000		EXIT

/LTR AC RWD (VIA MAINT GATE) FAILED
 /AC RWD

/DATA SENT FROM AC
 /DATA RECEIVED FROM RWD

6367	0014	TBRWBA,	0014
6370	2422		2422
6371	4001		4001
6372	0340		0340
6373	2227		2227
6374	0240		0240
6375	5026		5026
6376	1101		1101
6377	4024		4024
6400	0251		0251
6401	4006		4006
6402	0111		0111
6403	1405		1405
6404	0400		0400
6405	0103		0103
6406	4040		4040
6407	4022		4022
6410	2702		2702
6411	4000		4000
6412	7777		EXITA
6413	0005		REGB
6414	0006		REGC
6415	0000		EXIT
6416	0014	MAINTX,	0014
6417	2415		2415
6420	2240		2240
6421	1122		1122
6422	4022		4022
6423	0507		0507
6424	4006		4006
6425	0111		0111
6426	1405		1405
6427	0440		0440
6430	2417		2417
6431	4023		4023
6432	0524		0524
6433	4000		4000
6434	7777		EXITA
6435	0005		REGB
6436	0000		EXIT
6437	0014	LCXRB1,	0014
6440	0330		0330
6441	4005		4005
6442	3024		3024
6443	0516		0516
6444	0440		0440
6445	1720		1720
6446	2340		2340
6447	2205		2205
6450	0104		0104
6451	4002		4002
6452	0103		0103
6453	1300		1300
6454	7777		EXITA
6455	0004		REGA

/LTR AC RWB (VIA TB) FAILED
/AC RWB

/DATA SENT FROM AC
/DATA RECEIVED FROM RWB

/LTMR IR REG FAILED TO SET

/DATA RECEIVED WITH (17)

/LCX EXTEND OPS READ BACK
/0000 0000

/DATA SEND

6456	0005	REGB
6457	0000	EXIT
6460	0014	TAG, 0014
6461	0330	0330
6462	4014	4014
6463	0406	0406
6464	4060	4060
6465	6340	6340
6466	0401	0401
6467	2401	2401
6470	4030	4030
6471	0605	0605
6472	2240	2240
6473	0601	0601
6474	1114	1114
6475	0504	0504
6476	0004	0004
6477	0640	0640
6500	4040	4040
6501	2404	2404
6502	0600	0600
6503	7777	EXITA
6504	0006	REGC
6505	0007	REGD
6506	0000	EXIT
6507	0014	TAGA, 0014
6510	0330	0330
6511	4014	4014
6512	1106	1106
6513	4004	4004
6514	0124	0124
6515	0140	0140
6516	3006	3006
6517	0522	0522
6520	4006	4006
6521	0111	0111
6522	1405	1405
6523	0400	0400
6524	1106	1106
6525	4040	4040
6526	4024	4024
6527	1106	1106
6530	4000	4000
6531	7777	EXITA
6532	0006	REGC
6533	0007	REGD
6534	0000	EXIT

/DATA RECEIVED

/LCX LDF 03 DATA XPIR FAILED

/DF TDF

/0000 0000

/DATA SENT

/DATA RXED

/LCX LIF DATA XFER FAILED

/IF TIF

/0000 0000

/DATA SENT IF

/DATA RXED TIF

6535	0014	UNONH.	0014	/LTC UNIT 1 FAILED
6536	2403		2403	/0000
6537	4025		4025	
6540	1611		1611	
6541	2440		2440	
6542	6140		6140	
6543	0601		0601	
6544	1114		1114	
6545	0504		0504	
6546	4000		4000	
6547	7777		EXITA	
6550	0005		REGB	/UNIT SELECT DATA
6551	0000		EXIT	
6552	0014	WRENB.	0014	/LTC WRITE ENABLE FAILED
6553	2403		2403	/0000
6554	4027		4027	
6555	2211		2211	
6556	2405		2405	
6557	4005		4005	
6560	1601		1601	
6561	0214		0214	
6562	0540		0540	
6563	0601		0601	
6564	1114		1114	
6565	0504		0504	
6566	4000		4000	
6567	7777		EXITA	
6570	0000		EXIT	
6571	0014	UNSELM.	0014	/LTC UNIT SEL FAILED
6572	2403		2403	/0000
6573	4025		4025	
6574	1611		1611	
6575	2440		2440	
6576	2305		2305	
6577	1440		1440	
6600	0601		0601	
6601	1114		1114	
6602	0504		0504	
6603	4000		4000	
6604	7777		EXITA	
6605	0005		REGB	/SELECT DATA
6606	0000		EXIT	

6607	0014	GPERS,	0014	/LGP COUNT FAILED
6610	0720		0720	/0000 0000
6611	4003		4003	
6612	1725		1725	
6613	1624		1624	
6614	4006		4006	
6615	0111		0111	
6616	1405		1405	
6617	0400		0400	
6620	7777		EXITA	
6621	0005		REGB	/GOOD
6622	0006		REGC	/BAD
6623	0000		EXIT	
6624	0014	TFM1,	0014	/LTD TAPE FAIL DELAY FAILED TP0
6625	2402		2402	
6626	4024		4024	
6627	0120		0120	
6630	0540		0540	
6631	0601		0601	
6632	1114		1114	
6633	4004		4004	
6634	0514		0514	
6635	0131		0131	
6636	4006		4006	
6637	0111		0111	
6640	1405		1405	
6641	0440		0440	
6642	2420		2420	
6643	6000		6000	
6644	0000		EXIT	
6645	0014	TFM2,	0014	/LTD TAPE FAIL MAINT 0 IN PROG 1
6646	2404		2404	
6647	4024		4024	
6650	0120		0120	
6651	0540		0540	
6652	0601		0601	
6653	1114		1114	
6654	4015		4015	
6655	0111		0111	
6656	1624		1624	
6657	4060		4060	
6660	4011		4011	
6661	1640		1640	
6662	2022		2022	
6663	1707		1707	
6664	4061		4061	
6665	0000		EXIT	
6666	0014	TFM3,	0014	/LTD TAPE FAIL MAINT 1 IN PROG 1
6667	2404		2404	
6670	4024		4024	
6671	0120		0120	
6672	0540		0540	

6673	0601	0601
6674	1114	1114
6675	4015	4015
6676	0111	0111
6677	1624	1624
6700	4061	4061
6701	4011	4011
6702	1640	1640
6703	2022	2022
6704	1707	1707
6705	4061	4061
6706	0000	EXIT

6707	0014	TFDM3, 0014
6710	2404	2404
6711	4204	4204
6712	0120	0120
6713	0540	0540
6714	0601	0601
6715	1114	1114
6716	4015	4015
6717	0111	0111
6720	1624	1624
6721	4060	4060
6722	4011	4011
6723	1640	1640
6724	2022	2022
6725	1707	1707
6726	4060	4060
6727	0000	EXIT

6730	0014	AC1PT1, 0014
6731	2404	2404
6732	0103	0103
6733	1120	1120
6734	4050	4050
6735	0411	0411
6736	2205	2205
6737	0324	0324
6740	1117	1117
6741	1650	1650
6742	6151	6151
6743	5140	5140
6744	0601	0601
6745	1114	1114
6746	0504	0504
6747	0000	EXIT

6750	0014	AC1PT2, 0014
6751	2404	2404
6752	4001	4001
6753	0311	0311
6754	2040	2040
6755	5004	5004
6756	1122	1122

/LTD TAPE FAIL MAINT 0 IN PROG 0

/LTD ACIP (DIRECTION (1)) FAILED

/LTD ACIP (DIRECTION (0)) FAILED

6757	0503	0503
6760	2411	2411
6761	1716	1716
6762	5061	5061
6763	5151	5151
6764	4006	4006
6765	0111	0111
6766	1405	1405
6767	0440	0440
6770	0000	EXIT

6771	0014	0014
6772	2404	2404
6773	4001	4001
6774	0311	0311
6775	2040	2040
6776	5015	5015
6777	1724	1724
7000	1117	1117
7001	1640	1640
7002	5061	5061
7003	5151	5151
7004	4005	4005
7005	0111	0111
7006	1405	1405
7007	0400	0400
7010	0000	EXIT

ACIPT3,

/LTD ACIP (MOTION (1)) FAILED

7011	0014	0014
7012	1525	1525
7013	4004	4004
7014	1122	1122
7015	0503	0503
7016	2411	2411
7017	1716	1716
7020	4006	4006
7021	1417	1417
7022	2050	2050
7023	2305	2305
7024	2440	2440
7025	0617	0617
7026	2227	2227
7027	0122	0122
7030	0451	0451
7031	4000	4000
7032	0000	EXIT

DIRM1,

/LMU DIRECTION FLOP (SET FORWARD)

7033	0014	0014
7034	1525	1525
7035	4004	4004
7036	1122	1122
7037	0503	0503
7040	2411	2411
7041	1716	1716
7042	4006	4006

DIRM2,

/LMU DIRECTION FLOP (SET REVERSE)

7043	1417	1417
7044	2050	2050
7045	2305	2305
7046	2440	2440
7047	2205	2205
7050	2605	2605
7051	2223	2223
7052	0551	0551
7053	4000	4000
7054	0000	EXIT

7055	0014	DIRM3,	0014
7056	1525		1525
7057	4004		4004
7060	1122		1122
7061	0503		0503
7062	2411		2411
7063	1716		1716
7064	4006		4006
7065	1417		1417
7066	2040		2040
7067	5023		5023
7070	0524		0524
7071	4022		4022
7072	0526		0526
7073	0522		0522
7074	2305		2305
7075	5140		5140
7076	1517		1517
7077	2411		2411
7100	1716		1716
7101	4060		4060
7102	4004		4004
7103	1431		1431
7104	4063		4063
7105	4000		4000
7106	0000		EXIT

/LMU DIRECTION FLOP (SET REVERSE) MOTION 0 DLX3

7107	0014	DIRM4,	0014
7110	1525		1525
7111	4004		4004
7112	1122		1122
7113	0503		0503
7114	2411		2411
7115	1716		1716
7116	4006		4006
7117	1417		1417
7120	2040		2040
7121	5023		5023
7122	0524		0524
7123	4022		4022
7124	0526		0526
7125	0522		0522
7126	2305		2305
7127	5140		5140
7130	1561		1561
7131	6167		6167
7132	4003		4003
7133	6263		6263
7134	4000		4000
7135	0000		EXIT

/LMU DIRECTION FLOP (SET REVERSE) M117 C23

7136	0014	DIRM5,	0014
7137	1525		1525
7140	4004		4004

/LMU DIRECTION FLOP M117 C23 FALSE COND

7141	1122	1122
7142	0503	0503
7143	2411	2411
7144	1716	1716
7145	4006	4006
7146	1417	1417
7147	2040	2040
7150	1561	1561
7151	6167	6167
7152	4003	4003
7153	6263	6263
7154	4006	4006
7155	0114	0114
7156	2305	2305
7157	4003	4003
7160	1716	1716
7161	0411	0411
7162	2411	2411
7163	1716	1716
7164	2300	2300
7165	0000	EXIT

7166	0014	DIRM6,	0014
7167	1525		1525
7170	4004		4004
7171	1122		1122
7172	0503		0503
7173	2411		2411
7174	1716		1716
7175	4006		4006
7176	1417		1417
7177	2040		2040
7200	1561		1561
7201	6165		6165
7202	4003		4003
7203	6267		6267
7204	4015		4015
7205	2402		2402
7206	4000		4000
7207	0000		EXIT
7210	7777		EXITA
7211	0014	DIRM7,	0014
7212	1525		1525
7213	4004		4004
7214	1122		1122
7215	0503		0503
7216	2411		2411
7217	1716		1716
7220	4006		4006
7221	1417		1417
7222	2040		2040

/LMU DIRECTION FLOP M115 C27, MTB

/LMU DIRECTION FLOP M115 C27 TAC10(1)

7223	1561	1561
7224	6165	6165
7225	4003	4003
7226	6267	6267
7227	4024	4024
7230	0103	0103
7231	4061	4061
7232	6030	6030
7233	6151	6151
7234	4000	4000
7235	0000	EXIT
7236	0014	DIRMS, 0014
7237	1525	1525
7240	4004	4004
7241	1122	1122
7242	0303	0303
7243	2411	2411
7244	1716	1716
7245	4006	4006
7246	1417	1417
7247	2040	2040
7250	1561	1561
7251	6167	6167
7252	4002	4002
7253	6303	6303
7254	4006	4006
7255	0111	0111
7256	1405	1405
7257	0400	0400
7260	0000	EXIT

/LMU DIRECTION FLOP M117 1333 FAILED

7261	0014	TBX1,	0014
7262	2205		2205
7263	4015		4015
7264	0240		0240
7265	2402		2402
7266	1640		1640
7267	0601		0601
7270	1114		1114
7271	0504		0504
7272	4050		4050
7273	0530		0530
7274	2440		2440
7275	0104		0104
7276	0475		0475
7277	6040		6040
7300	5100		5100
7301	7777		EXITA
7302	0005		REGS
7303	0006		REGC
7304	0000		EXIT

/LRE MB TBN FAILED (EXT ADD = 0)

7305	0014	TBX2.	0014
7306	2205		2205
7307	4015		4015
7310	0240		0240
7311	2402		2402
7312	1640		1640
7313	0601		0601
7314	1114		1114
7315	0504		0504
7316	4050		4050
7317	0530		0530
7320	2440		2440
7321	0104		0104
7322	0475		0475
7323	6140		6140
7324	5100		5100
7325	7777		EXITA
7326	0009		REOB
7327	0006		REGC
7330	0000		EXIT

/LRE MB TON FAILED (EXT ADD = 1)
/0000 0000

7331	0014	LGM,	0014	/LWN LCB1 INHIBIT DECODING FAILED
7332	2716		2716	/0000
7333	4014		4014	
7334	0360		0360	
7335	6140		6140	
7336	1116		1116	
7337	1011		1011	
7340	0211		0211	
7341	2440		2440	
7342	0417		0417	
7343	0317		0317	
7344	0411		0411	
7345	1607		1607	
7346	4006		4006	
7347	0111		0111	
7350	1405		1405	
7351	0400		0400	
7352	7777		EXITA	
7353	0005		REGB	/DATA RECEIVED
7354	0000		EXIT	
7355	0014	LGM000,	0014	/LGP GP EQ GPC FAILED
7356	0720		0720	/0000
7357	4007		4007	
7360	2040		2040	
7361	0521		0521	
7362	4007		4007	
7363	2003		2003	
7364	4006		4006	
7365	0111		0111	
7366	1405		1405	
7367	0400		0400	
7370	7777		EXITA	
7371	0005		REGB	
7372	0000		EXIT	
7373	0014	LGM001,	0014	/LGP COMPARE FAILED
7374	0720		0720	/0000 0000
7375	4003		4003	
7376	1715		1715	
7377	2001		2001	
7400	2205		2205	
7401	4006		4006	
7402	0111		0111	
7403	1405		1405	
7404	0400		0400	
7405	7777		EXITA	
7406	0005		REGB	
7407	0006		REGC	
7410	0000		EXIT	
7411	0014	LGM003,	0014	/LGP FAILED TO COUNT M15 C16
7412	0720		0720	
7413	4006		4006	
7414	0111		0111	

7415	1405	1405
7416	0440	0440
7417	2417	2417
7420	4003	4003
7421	1725	1725
7422	1624	1624
7423	4015	4015
7424	6161	6161
7425	6340	6340
7426	0361	0361
7427	6600	6600
7430	0000	EXIT
7431	0014	0014
7432	0715	0715
7433	4015	4015
7434	6161	6161
7435	6740	6740
7436	0263	0263
7437	6540	6540
7440	0601	0601
7441	1114	1114
7442	0504	0504
7443	4024	4024
7444	1740	1740
7445	0314	0314
7446	0501	0501
7447	2240	2240
7450	0317	0317
7451	2516	2516
7452	2405	2405
7453	2200	2200
7454	0000	EXIT

LGM004,

/LGM M117 B35 FAILED TO CLEAR COUNTER

7455	0000	KTAC,
7456	7300	
7457	1016	
7460	6151	
7461	7200	
7462	6154	
7463	2255	
7464	5655	
7465	6141	LOCD,
7466	0700	
7467	0000	
7470	0002	
7471	5540	

0000	
CLA CLL	
TAD	K1400
LMR	
CLA	
XFR	
ISE	KTAC
JMP I	KTAC
LINC	
0700	
0000	
PDP	
JMP I	PNTE
S	

/GO LINC MODE

/B MODE
/EXIT

ACIP1	4017	ERROR	0021	K0600	0054	LDFK	0066
ACIP2	4034	ERRORS	4621	K0700	0014	LDPT0	2101
ACIP3	4045	EXAD1	1634	K0777	0073	LGB002	3722
ACIPT1	6730	EXIT	0000	K1000	0062	LGM000	7355
ACIPT2	6750	EXITA	7777	K1007	0143	LGM001	7373
ACIPT3	6771	FINAL	5343	K1026	0041	LGM003	7411
ADDS1	1501	FMTST	3206	K1400	0016	LGM004	7431
ADDS2	1560	GETWIN	0107	K2000	0035	LGP000	3560
AOX	0001	GMTST	3142	K2007	0144	LGP001	3604
ASCIi	4654	GO	0176	K240	0040	LGP002	3641
ASCRA	0101	GPCNT1	3322	K2400	0141	LIPT0	2152
ASCRXT	4627	GPCNT2	3357	K3000	0002	LIN01	2235
BADEXIT	5163	GPCNT3	3403	K3400	0047	LIN02	2261
BELL	0130	GPCNT5	3475	K4000	0150	LIN04	2311
BELLS	5011	GPCNT6	3530	K4400	0036	LIN05	2345
BLOCK	5352	GPCNTR	3441	K4440	0122	LIN06	2410
BMTST	3230	GPEDB	6127	K5000	0037	LIN07	2453
C0040	0170	GPERS	6607	K5040	0146	LIN77	5353
C7773	0151	GPER6	6176	K5252	0133	LINC	6141
CHECK	5316	GPM0	6163	K5400	0015	LINIDT	5631
CLR	0011	GPPRE	6150	K6000	0045	LININD	5610
CMTST	3120	GPSET	0147	K6040	0124	LINIPS	5571
CRLF	4753	GPSETS	5244	K7000	0003	LMR	6151
CRLFA	0171	GUARD	5325	K7030	0071	LMRWB	0064
DATA	5334	HERE	4736	K7356	0142	LMRWS	5065
DATABD	5766	IMTST	3252	K7400	0031	LOAD	0001
DATAPW	6006	INTERM	5361	K7760	0044	LOADR	0157
DATUM	4720	K0001	0116	K7761	0060	LOADS	4400
DATUMA	0100	K0002	0117	K7762	0127	LOCA	0152
DATYP	4762	K0003	0065	K7763	0126	LOGB	2230
DATYPA	0173	K0004	0046	K7767	0125	LOGC	5060
DIRM1	7011	K0006	0131	K7773	0061	LOGD	7465
DIRM2	7033	K0007	0070	K7774	0123	LSTERR	0033
DIRM3	7055	K0010	0043	K7776	0057	LWNDT1	2722
DIRM4	7107	K0013	0017	K7777	0115	LWNDT2	3035
DIRM5	7136	K0016	0121	K7777A	0074	LWNRN1	3004
DIRM6	7166	K0017	0013	KST	2174	LWNRN2	2754
DIRM7	7211	K0020	0051	KTAC	7455	M40	0106
DIRM8	7236	K0030	0075	KTACA	0102	MAINT1	0022
DIRT1	4061	K0040	0052	KX0010	0077	MAINT2	5540
DIRT2	4103	K0050	0076	LC010	3274	MAINT4	5462
DIRT3	4125	K0070	0072	LC0M	7331	MAINTC	5422
DIRT4	4154	K0077	0042	LCOM	6225	MAINTI	5445
DIRT5	4210	K0100	0053	LCX01	1712	MAINTM	5476
DIRT6	4246	K0137	0120	LCX02	1741	MAINTP	5515
DIRT7	4275	K0137A	0067	LCX03	1764	MAINTS	5000
DIRT8	4326	K0177	0056	LCX04	2012	MAINTX	6416
DISPAT	2167	K0200	0167	LCX05	2050	NERROR	0103
DMTST	3164	K0207	0105	LCXDT1	6256	NERR0S	4600
DOMORE	4406	K0212	0050	LCXLTD	6316	NOPL	0016
EMTST	3076	K0215	0055	LCXRB1	6437	OCTYP	4732
END	5370	K0400	0063	LCXTPS	6277	POP	0002

PINT	0012	TFDLY1	3727
PNTA	0134	TFDLY2	3743
PNTB	0135	TFDLY3	3763
PNTC	0136	TFDLY4	4003
PNTD	0137	TFDM3	6707
PNTE	0140	TFM1	6624
RAN	0032	TFM2	6645
RANDM	5022	TFM3	6666
REDO	4737	TFSUB	5200
REDOA	0172	TIME	5132
REGA	0004	TIMER	0132
REGB	0005	TIMTF	0145
REGC	0006	TMA	0023
REGD	0007	TMAAC1	1014
REGE	0010	TMAAC2	1043
REGF	0011	TMAC	5377
RNA	5047	TMASET	6026
RNB	5050	TMR01	0201
RNC	5051	TMR02	0226
RWBMG	6334	TMR03	0253
RWBMS1	1261	TMR04	0303
RWBMS2	1304	TMR05	0337
RWBSH1	1162	TMR06	0371
RWBSH2	1220	TMR07	0424
RWBSHF	6056	TPEPRE	0024
SKPL	0467	TRC	6152
SPACE	0034	TYPE	0023
TAC	0003	TYPECH	4703
TACAC	5707	TYPOT	5052
TACAC1	0620	UNIDAT	6072
TACAC2	0647	UNIDTA	2510
TAG	6460	UNIONE	2633
TAGA	6507	UNONM	6535
TASAC1	0735	UNSEL	2664
TASAC2	0762	UNSELM	6571
TBAC	5666	UNTMES	6110
TBAC1	0442	UNTPRE	2567
TBAC2	0467	WRENB	6552
TBADD	5731	WRTENB	2703
TBNAC1	0534	XPR	6154
TBNAC2	0563	XOA	0021
TBNMB1	1327		
TBNMB2	1361		
TBNMB3	1413		
TBNMB4	1445		
TBRWB1	1101		
TBRWB2	1130		
TBRWBA	6367		
TBWAC	5643		
TBX1	7261		
TBX2	7305		
TEMPB	0104		

ERRORS DETECTED: 0

LINKS GENERATED: 0

RUN-TIME: 33 SECONDS

3K CORE USED